\Orchestrating a brighter world NEC

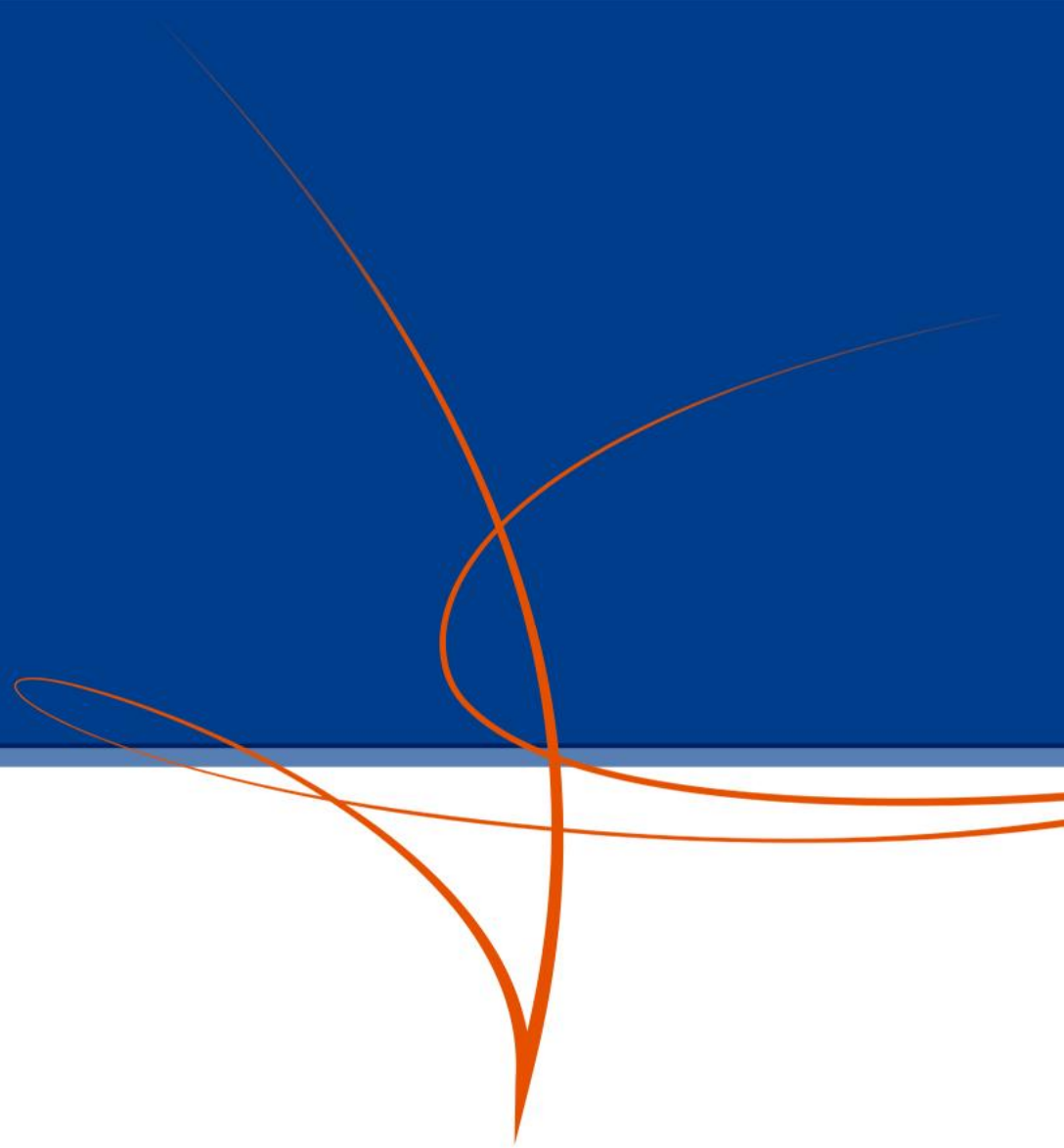
NEC SX-Aurora TSUBASA

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.

NEC HPC Business



NEC Business

NEC at a Glance

Establishment

Established

1899

Sales

Over

US$26B

in annual revenue (FY2018)*1

Patents

Approximately

49,000*3

AI Researchers

500

Tech Leadership

World **No. 1** *2

Face Recognition

Iris Recognition

Fingerprint Recognition

Anniversary

120

years

Thank you for all your support.
NEC is committed to being
a great partner and corporate citizen
for another 120 years to come.

Global Networks

327 group companies,

169 countries & regions

Employees

Over **110,000**

employees

*1 USD yen equivalent calculated at USD1=110.99 yen *2 According to tests performed by the U.S. National Institute of Standards and Technology

*3 Cumulative total, as of March 31, 2019

NEC Technology Labs

Where Ideas for the Future Are Born

NEC is working with customers and external partners worldwide to co-create a brighter future. These are some of our research laboratories and facilities for experiencing cutting-edge technologies and designing the future.

NEC Laboratories America
Development of cutting-edge core technologies

NEC Laboratories Europe
R&D through EU projects and standardization activities

Israel Research Center
Promote marketing-type research

NEC Laboratories India
Create core technologies to address the challenges of emerging countries

NEC Laboratories Japan
Takes the central role in NEC R&D, focusing on AI (recognition, analysis), security, ICT platform, quantum computing

NEC Laboratories China
Development of core technologies in AI and network

NEC Laboratories Singapore
Co-creation of innovative solutions with customers and partners, mainly for developed countries

NEC Future Creation Hub

NEC Future Creation Hub is where technology and business come together through interactive dialogue and directly experiencing cutting-edge technologies so as to co-create the future. Our experts work with global clients to tackle challenges head-on. Together, we develop initiatives that generate social value for the current and future generations.

NEC / Netcracker Innovation Center

The Innovation Center is a fully immersive, specially designed facility to help us foster personalized interactions with customers and partners, and promote co-creation. Here, partners and customers can experience in-depth demonstrations of our cutting-edge innovations in virtualization technologies, next-generation BiOSS and DX enablement capabilities, and together we can collaborate to develop customized solutions that meet their respective needs.

NEC Open Innovation Centre

NEC Open Innovation Centre is a business-driven showcase where visitors can experience NEC's trusted Digital ID to access government and commercial services. The Centre offers public safety solutions based on biometrics and artificial intelligence, providing unparalleled safety, security and seamlessness within the context of daily life. We are ready to conduct proof of concept demonstrations of these solutions with our partners and customers.

Solution Areas in EMEA





Carrier

Transport solutions *(incl. mobile backhaul)*

Small cells as a Service (ScaaS)

Emergency service solutions

SDN/NFV

TOMS (OSS/BSS)

Cloud services platform

Value added services

SDN/NFV solutions

Enterprise

Business mobility

Cloud

Unified communications & collaboration

High availability solutions

IT solutions
- *Retail*
- *Manufacturing*

Public sector

Public Safety

- *Face & fingerprint recognition*

- *Urban surveillance (cameras and analytics)*

- *Cyber security*

Broadcast

Smart cities, transport

Smart Energy

Energy Storage Systems (ESS)

Generation Energy Management Systems (GEMS)

Distributed & Storage Energy Management Systems (DEMS/SEMS)

Demand Response Systems (DREMS)

System platform

Desktop displays

Large format displays

Projectors

Digital cinema solutions

Supercomputers

Servers

Mainframes

Storage

NEC Our Global Scope

Business activities in over 160 countries and territories

Our affiliates, offices, and laboratories: 58 countries and territories

NEC HPC EMEA

NEC HPC portfolio

HPC Portfolio

SX–Aurora TSUBASA

Vector CPU

LX Series

X86 CPU & GPU

Storage

Scalable appliances

NEC Expertise

- Application tuning
- System architecting
- System integration
- Service & Support
- R&D

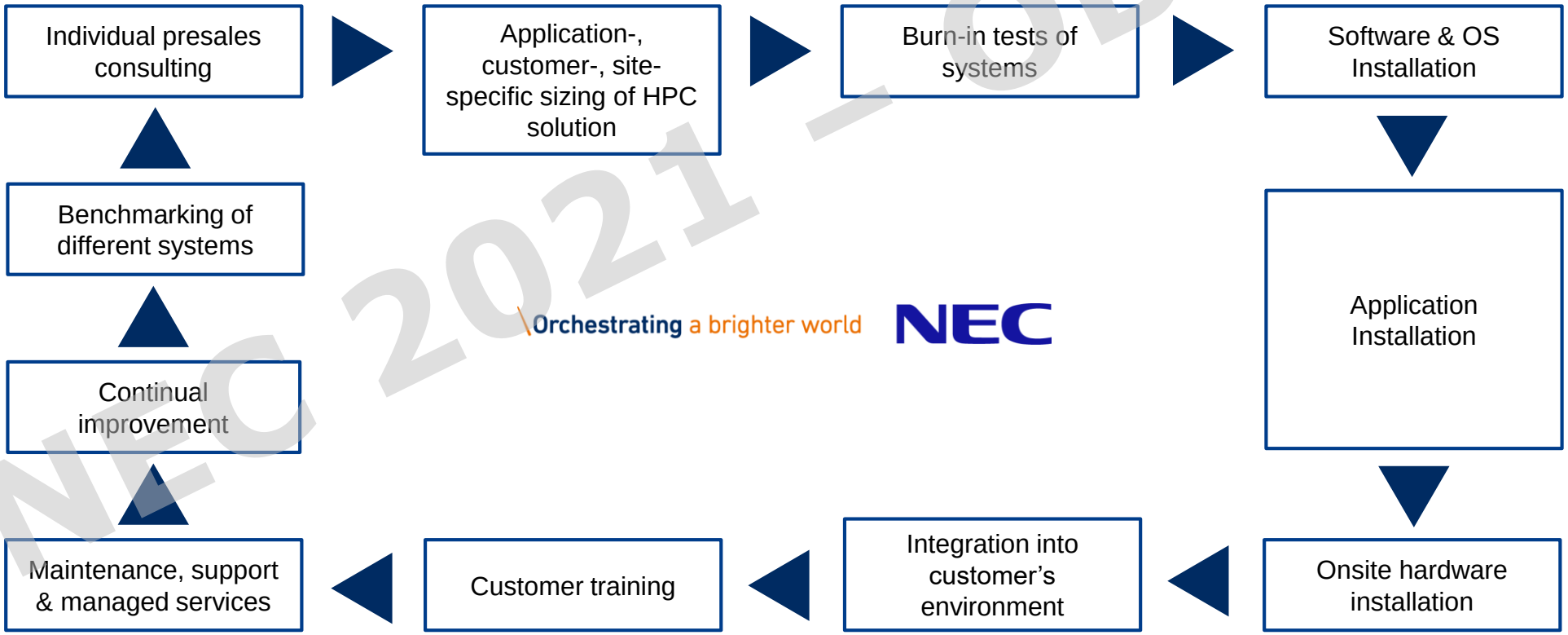
Sustained Performance




NEC Offers Direct Full Customer Service

We deliver **comprehensive solutions** for High Performance Computing (HPC)

- **Turnkey**
- Caring for the customer during the whole **solution lifecycle**:
  - From **individual consulting** to **managed services**
  - Customers have access to **NEC Aurora Benchmark Center**



NEC key strengths

NEC has expertise in software

- Software and compiler development
- Benchmarking and code optimization
- Cluster management solution development

NEC develops and builds optimized HPC hardware solutions

- SX vector architecture for maximizing sustained performance
- Storage appliances for parallel files systems

NEC services

- Users and system administrators trainings
- Onsite or remote administration
- Users application support (optimization)
- Workshops and hackathons
- Joint research activities

NEC – Main HPC Customers

Meteorology, Climatology, Environment

Industry



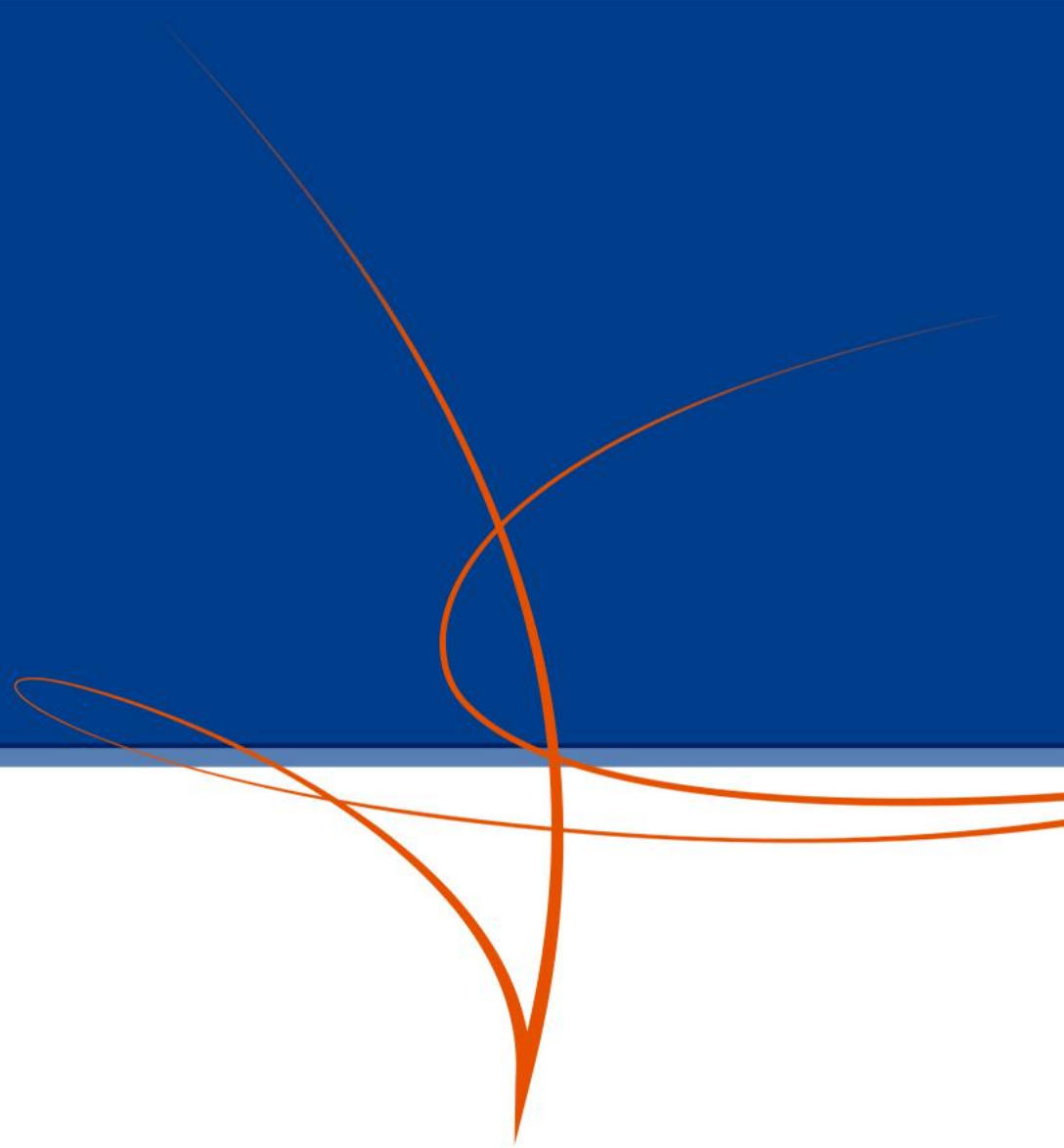

Research Centers

Universities, HPC Centers


NEC Top 500 entries as of November 2020

Rank	Site	System	Cores	Rmax (Tflop/s)	Rpeak (Tflop/s)
33	National Institute for Fusion Science (NIFS) Japan	Plasma Simulator - SX-Aurora TSUBASA A412-8, Vector Engine Type10AE 8C 1.58GHz, Infiniband HDR 200 NEC	34,560	7,892.7	10,510.70
140	Deutscher Wetterdienst Germany	SX-Aurora TSUBASA A412-8, Vector Engine Type10AE 8C 1.58GHz, InfiniBand HDR 100 NEC	14,848	2,595.6	4,360.00
152	Universitaet Aachen RWTH – IT Center Germany	CLAIX (2018) – Intel-HNS2600BPB, Xeon Platinum 8160 24C 2.1GHz, Intel Omni-Path NEC	61,200	2,483.60	4,112.60
251	Universitaet Mainz Germany	Mogon II – NEC Cluster, Xeon Gold 6130 16C 2.1 GHz, MEGWARE MiriQuid Xeon E5-2360v4, Intel Omni-Path NEC/MEGWARE	49,432	1,967.80	2,800.90
298	Institute for Molecular Science Japan	Molecular Simulator – NEC LX Cluster, Xeon Gold 6148/6154, Intel Omni-Path NEC	38,552	1,785.60	3,072.80
304	German Aerospace Center Germany	CARA – NEC LX Cluster, AMD Epyc 7601 32C 2.2GHz, InfiniBand HDR NEC	145,920	1,746.00	2,568.20
437	Center for Computational Sciences University of Tsukuba Japan	Cygnus – NEC LX Cluster, Xeon Gold 6126 12C 2.6GHz, NVIDIA Tesla V100, InfiniBand HDR100 NEC	27,520	1,582.00	2,399.70

NEC SX-Aurora TSUBASA



Orchestrating a brighter world NEC

Developing Super Computing technologies for 35 years

Development Philosophy

SX-Aurora TSUBASA

POINT 1 Memory Bandwidth

Highest memory bandwidth per processor
Highest memory bandwidth per core

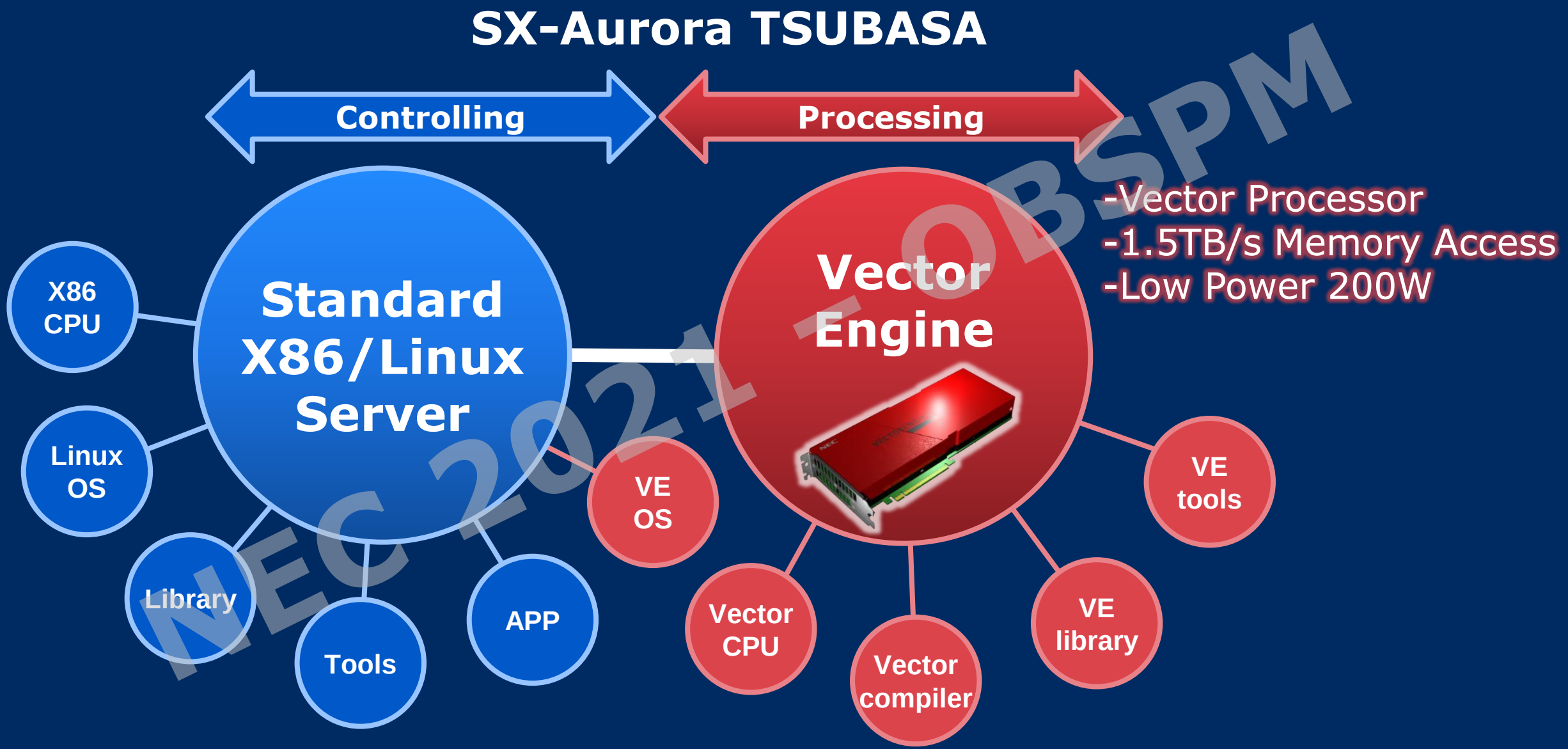
POINT 2 Easy to Use

Fortran/C/C++ programming, OpenMP
Automatic vectorization/parallelization

POINT 3 Energy Efficient

Power limitation for HPC systems becomes an issue
Higher sustained performance with lower power

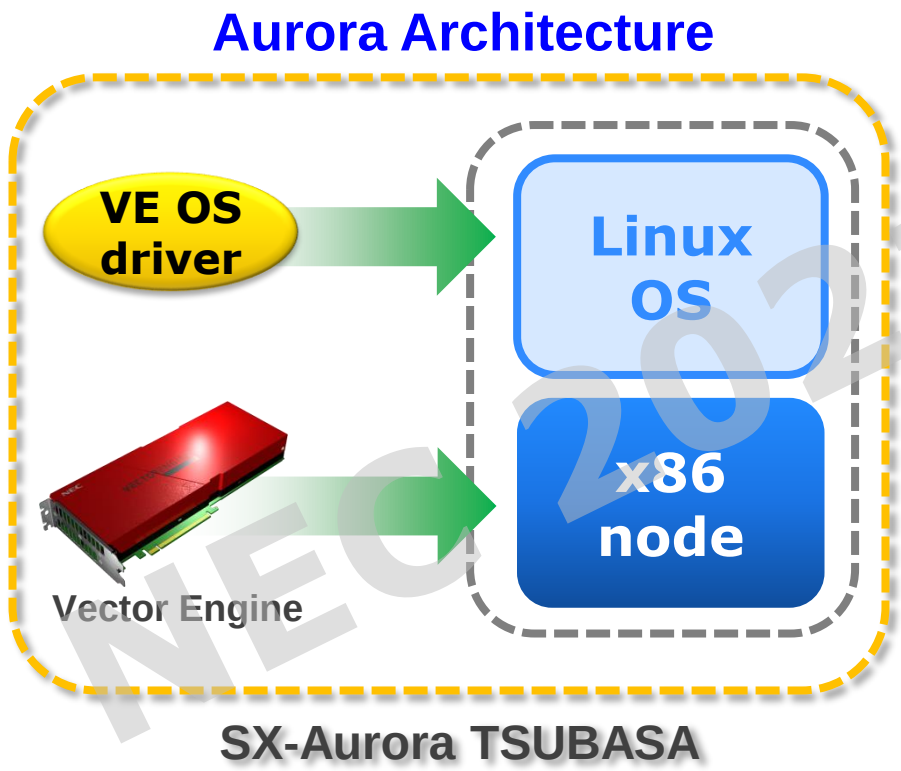
Architecture of SX-Aurora TSUBASA



Architecture

Aurora Architecture

- x86 node + Vector Engine (VE)
- VE capability is provided on x86/Linux environment



Product

- VE + x86 node

SW Environment

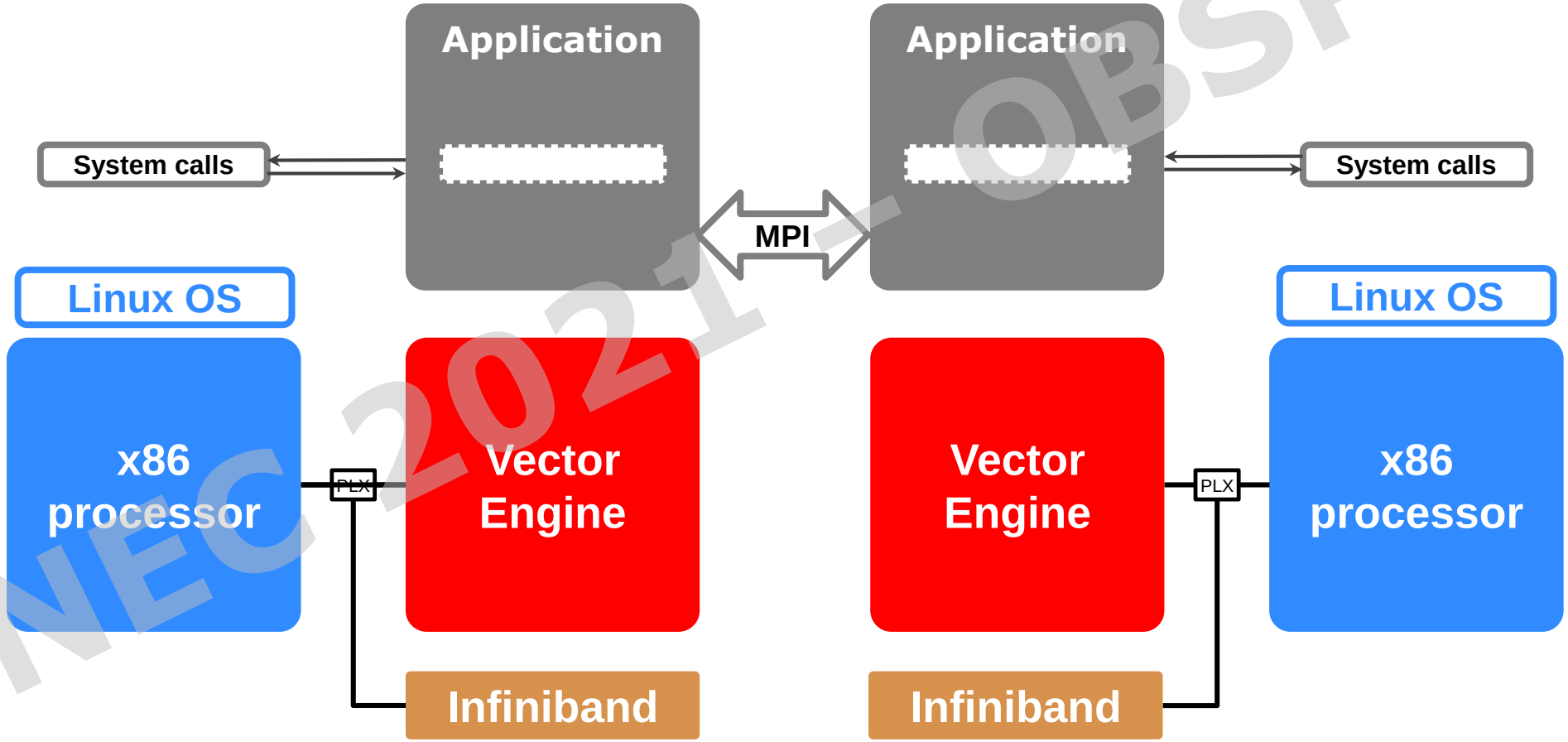
- x86 Linux OS
- Fortran/C/C++ standard programming
- Automatic vectorization by proven vector compiler

Interconnect

- InfiniBand for MPI

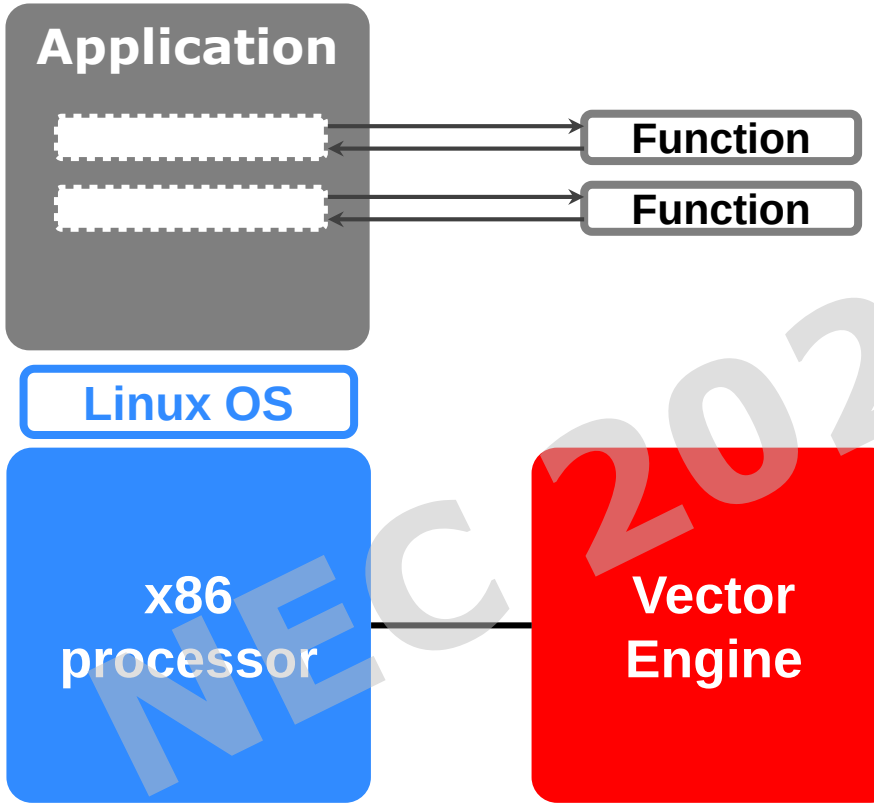
Execution modes - Native

- Vector Engine executes the entire application
- Only system calls (e.g. IOs) are handled by x86
- Direct MPI communications (x86 is not involved)

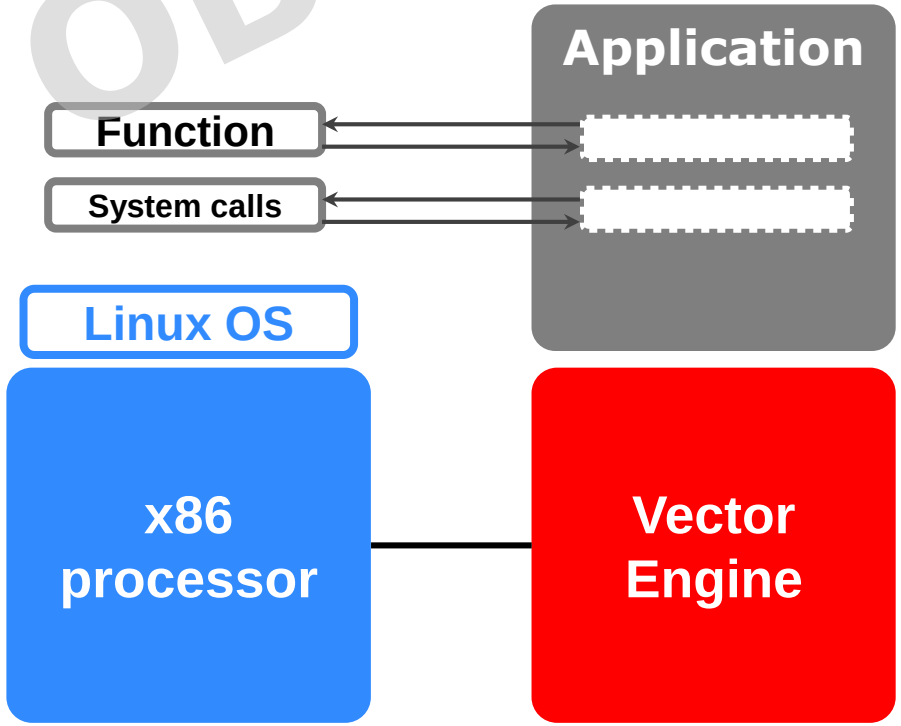


Execution modes - Offload

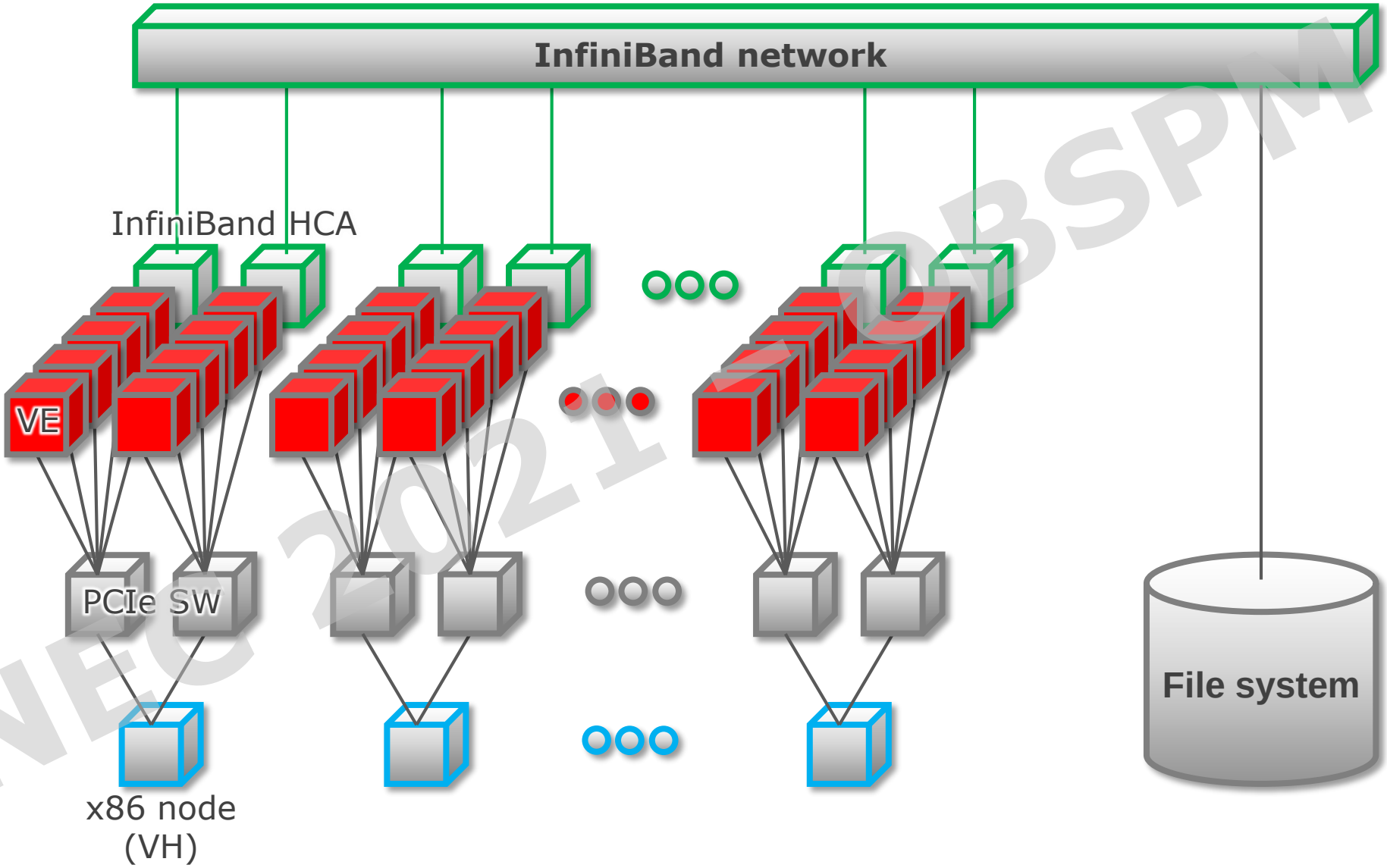
- Application runs on x86
- Offload vectorized functions to VE
- "Accelerator mode" such as GPU



- Application runs on Vector Engine
- Offload "inefficient" scalar function from VE to x86

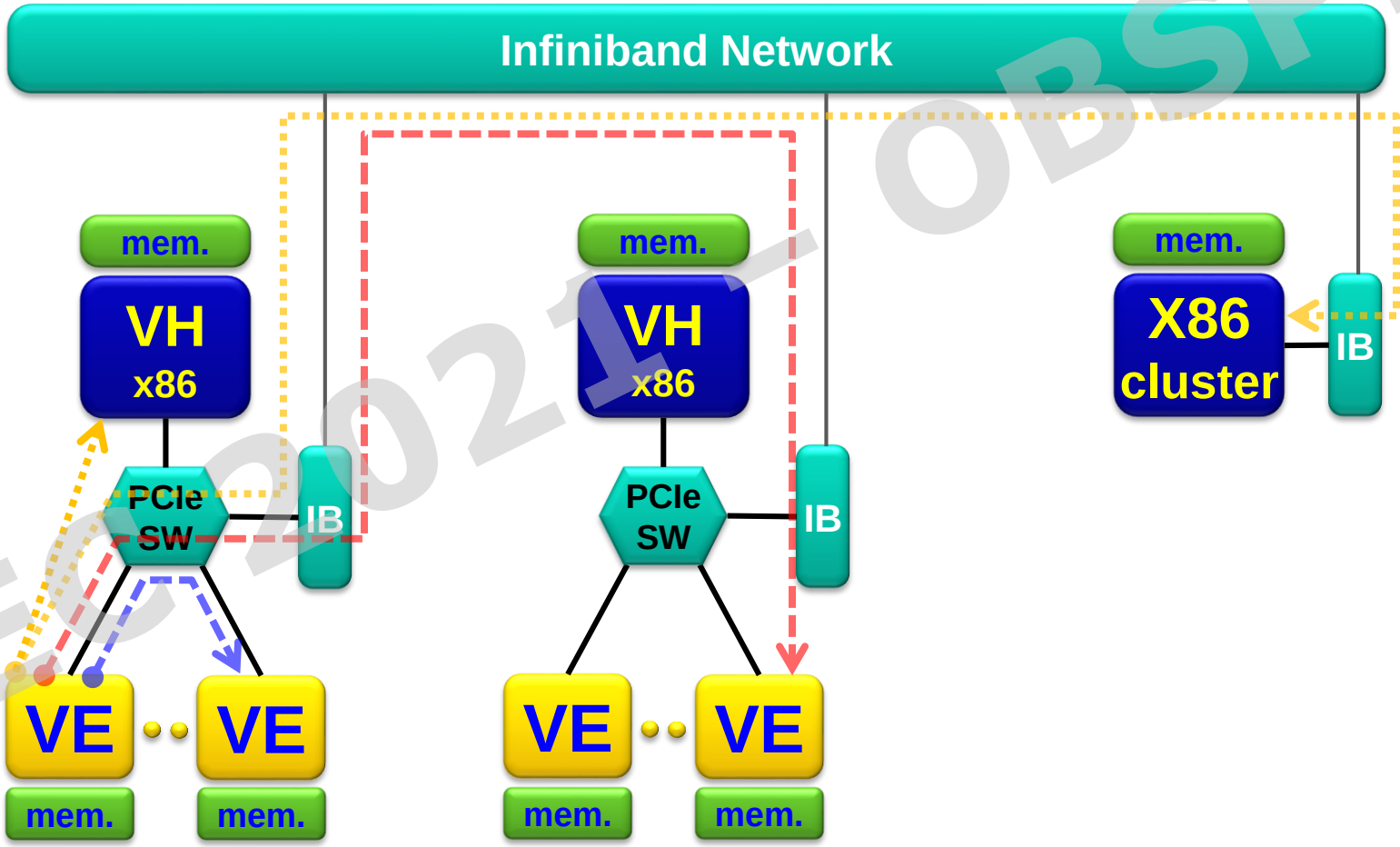


Large System Configuration



MPI communications

Direct communications between VEs (no x86 involved and RDMA)
Hybrid mode with processes on x86 & Aurora processors



Vector Engine (VE)

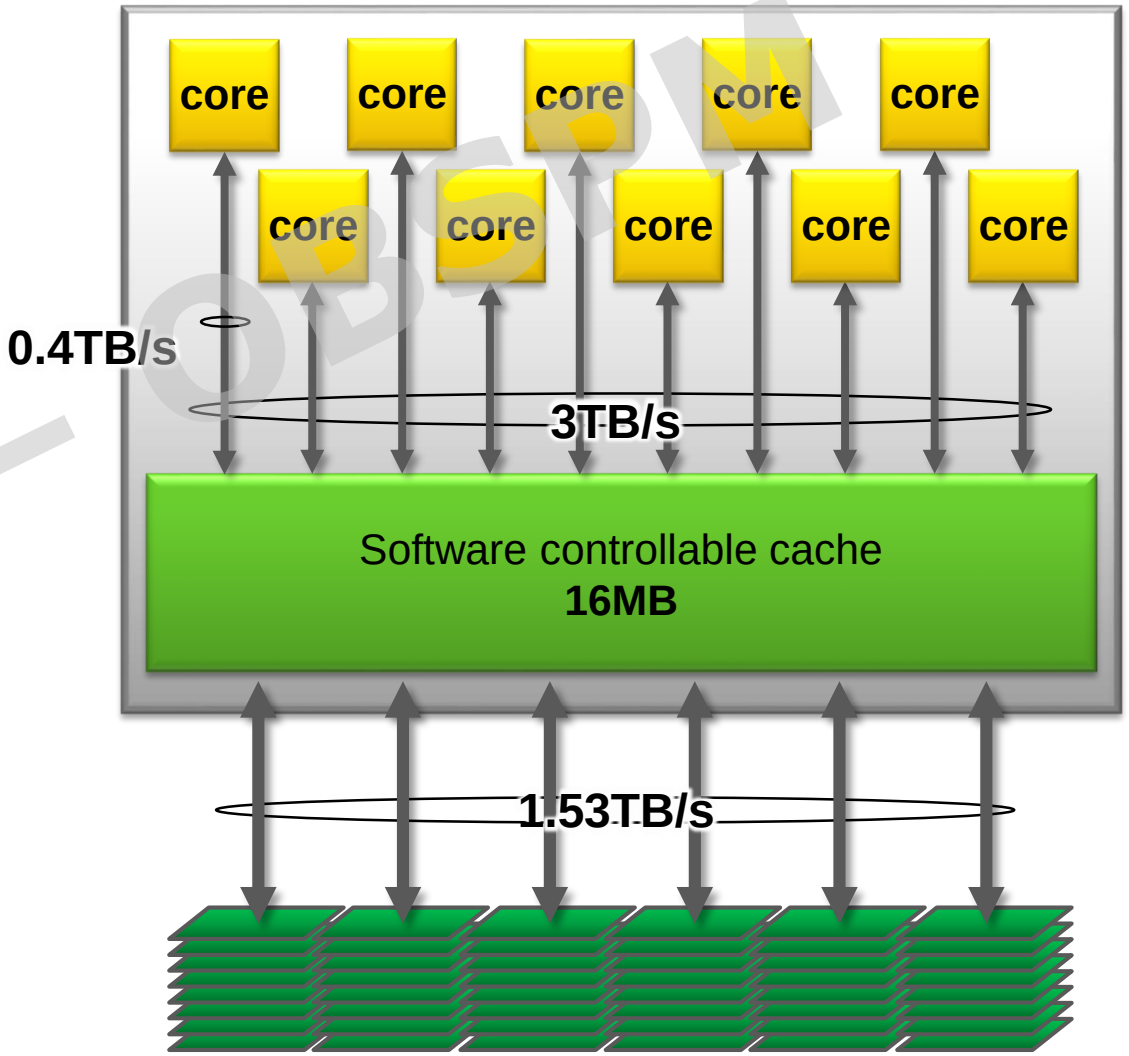
Card Implementation

- Standard PCIe implementation
- Connector: PCIe Gen.3 x16
- Double height (same form factor as Nvidia)
- <300W (DGEMM ~210W/VE, STREAM ~200W/VE, HPCG ~215W/VE)

VE20 Processor

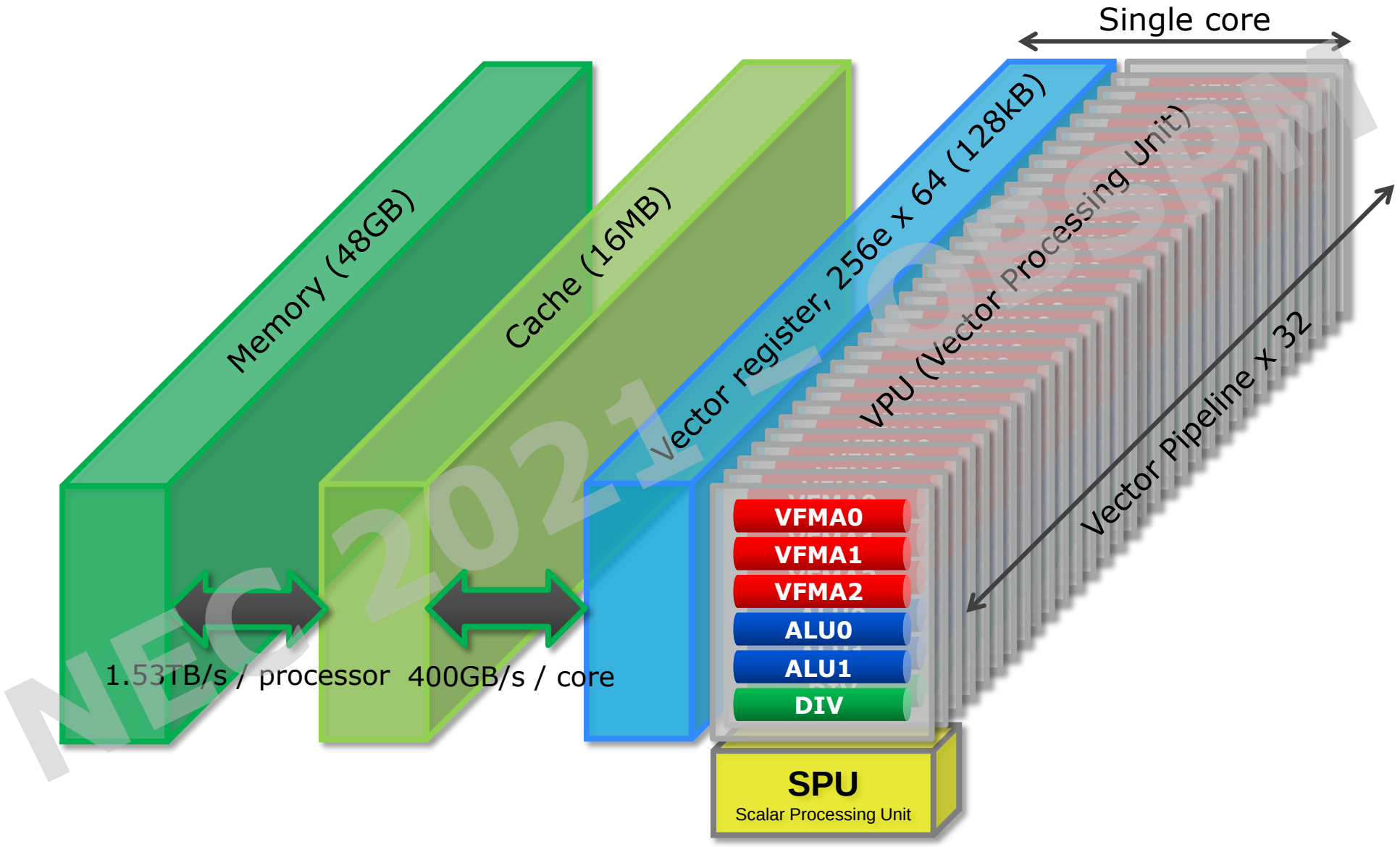
VE20 Specifications

Processor Version	Type A	Type B
Cores/processor	10	8
Core performance	307GF (DP) 614GF (SP)	
Processor performance	3.07TF (DP) 6.14TF (SP)	2.45TF (DP) 4.91TF (SP)
Cache capacity	16MB	
Cache bandwidth	3TB/s	
Cache Function	Software Controllable	
Memory capacity	48GB	
Memory bandwidth	1.53TB/s	
Power	~300W (TDP) ~200W (Application)	

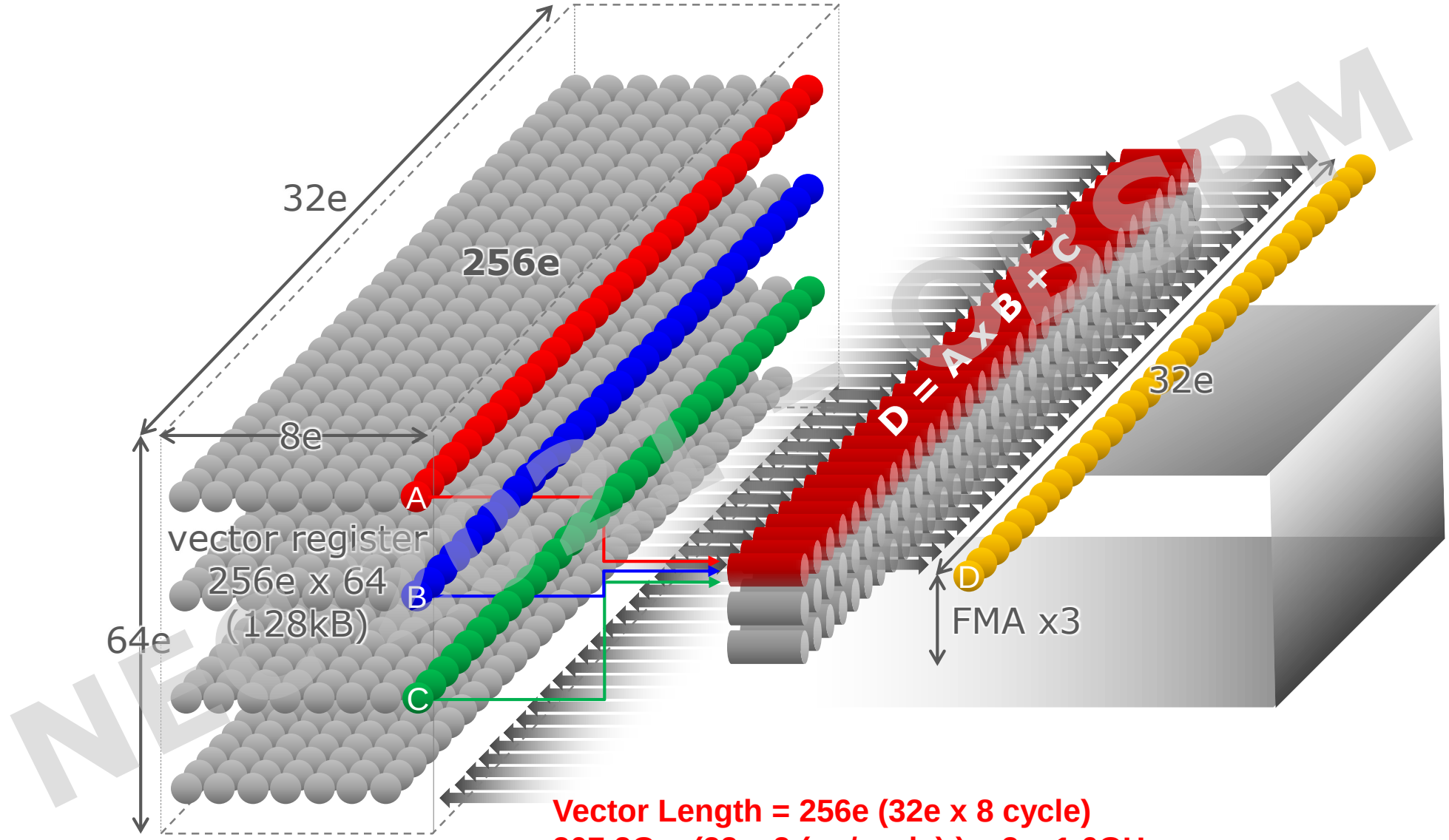


HBM2 memory x 6

Core Architecture

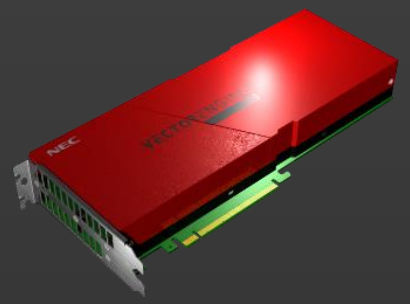


Vector Execution



Vector Length = 256e (32e x 8 cycle)
307.2G = (32 x 2 (op/cycle)) x 3 x 1.6GHz

VE Roadmap



Memory bandwidth / VE

VE10
8C/2.45TF
1.22TB/s memory bandwidth

VE10E
8C/2.45TF
1.35TB/s memory bandwidth

VE20
10C/3.07TF
1.5TB/s memory bandwidth

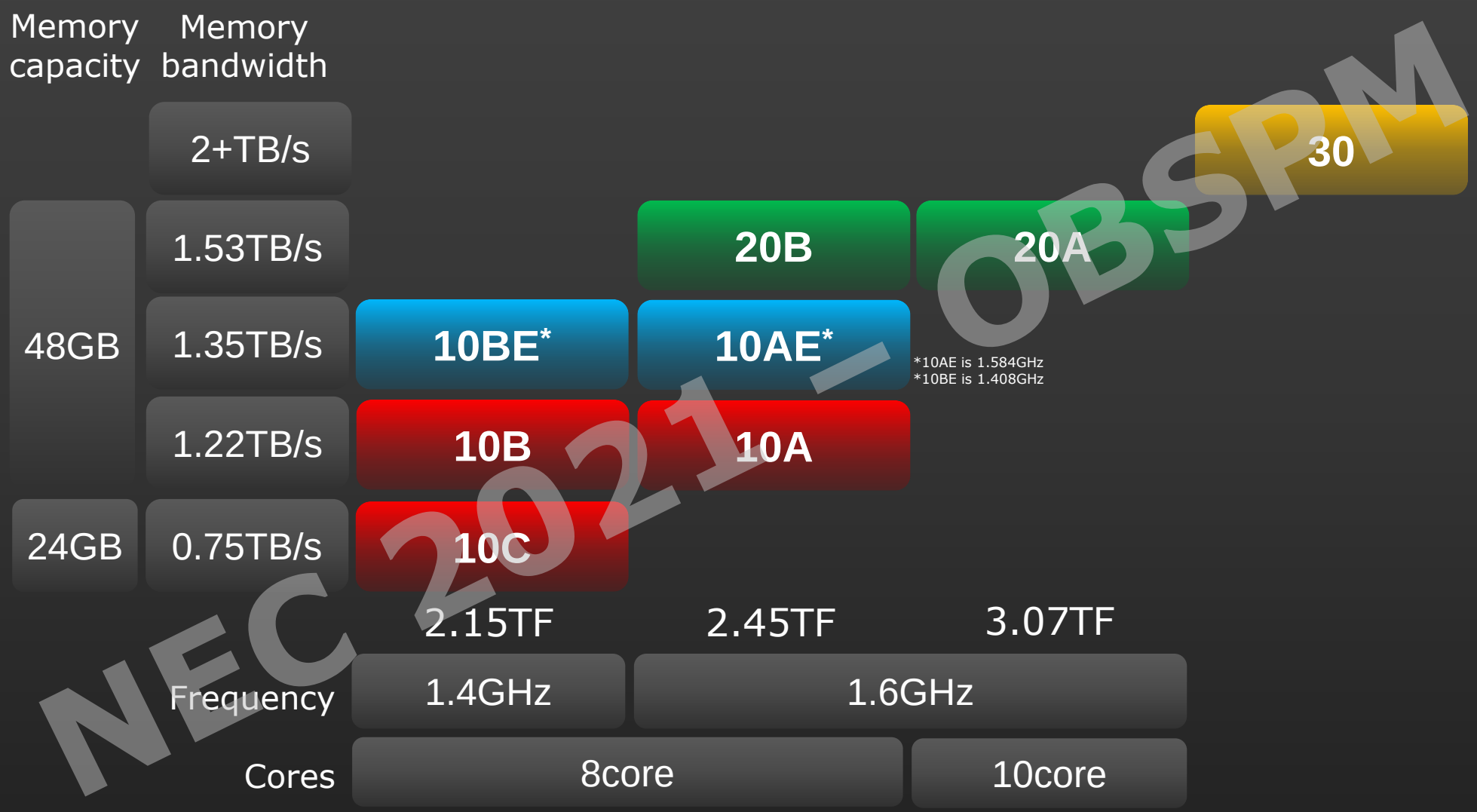
VE30
2+TB/s memory bandwidth

VE40
2++TB/s memory bandwidth

VE50

2019 2020 2021 2022 2023 2024

VE10/10E/20 SKU



Server products

Products

A500
A400

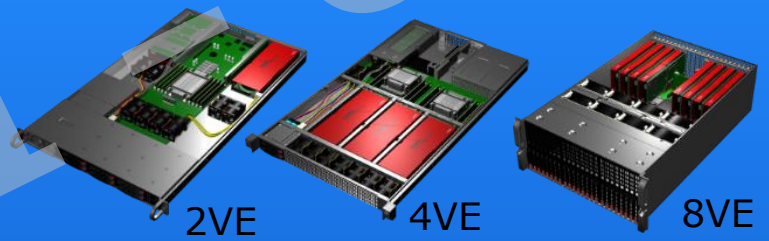
Supercomputer Model

- For large scale configuration
- DLC with 40C water

A300

Rack Mount Model

- Flexible configuration
- Air Cooled



A100

Tower Model

- For developer/programmer
- Tower implementation

B401-8 High Density DLC Model (Components)

Direct Liquid Cooling

VE DLC

To provide higher density
Hot water cooling (~45C)

8VE/2U

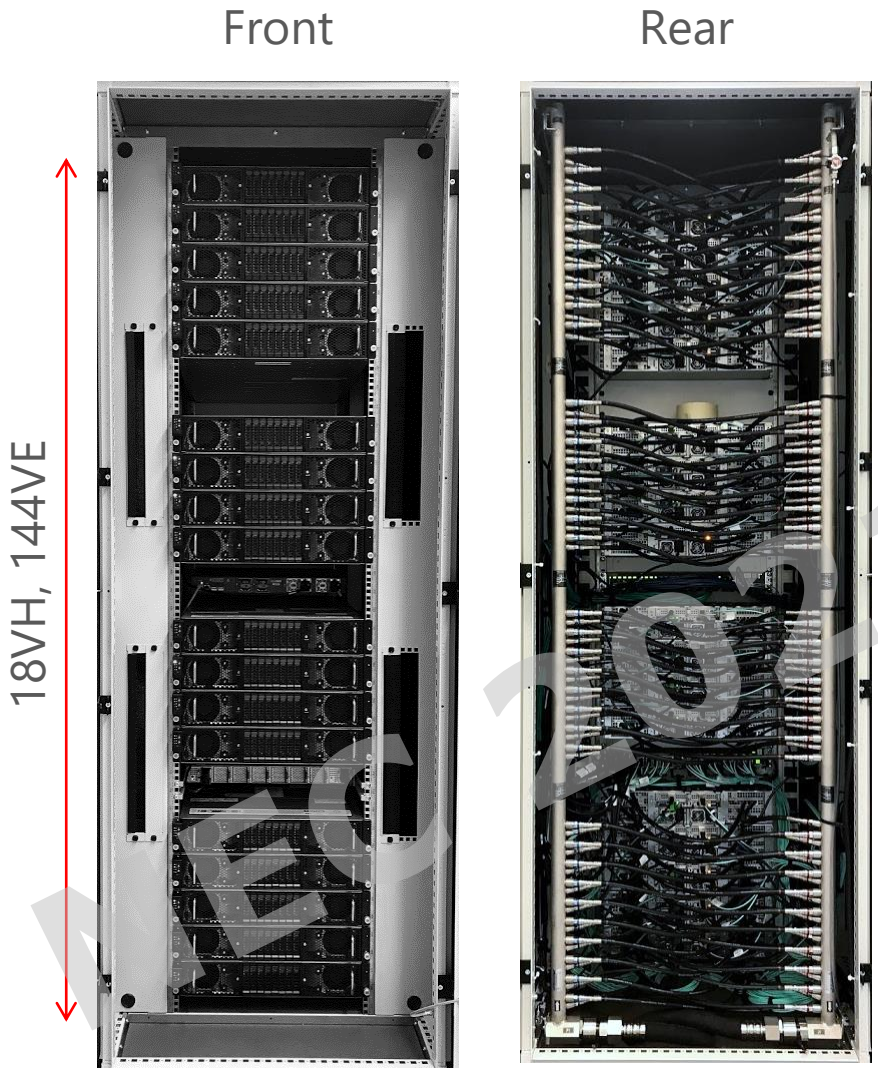
B401-8

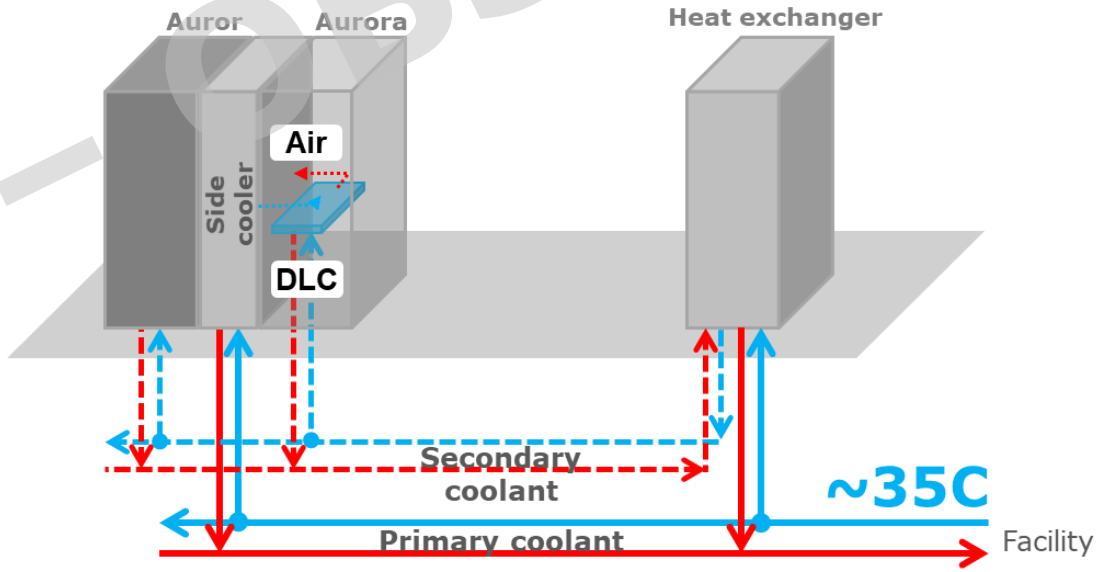
VE x8
AMD Rome processor x1
IB HCA x2

Performance:	24.5TF /VH
Memory bandwidth:	12.2TB/s /VH
HPL:	2.4kW
HPCG:	1.7kW

Up to 18VH, 144VE per rack

B401-8 High Density DLC

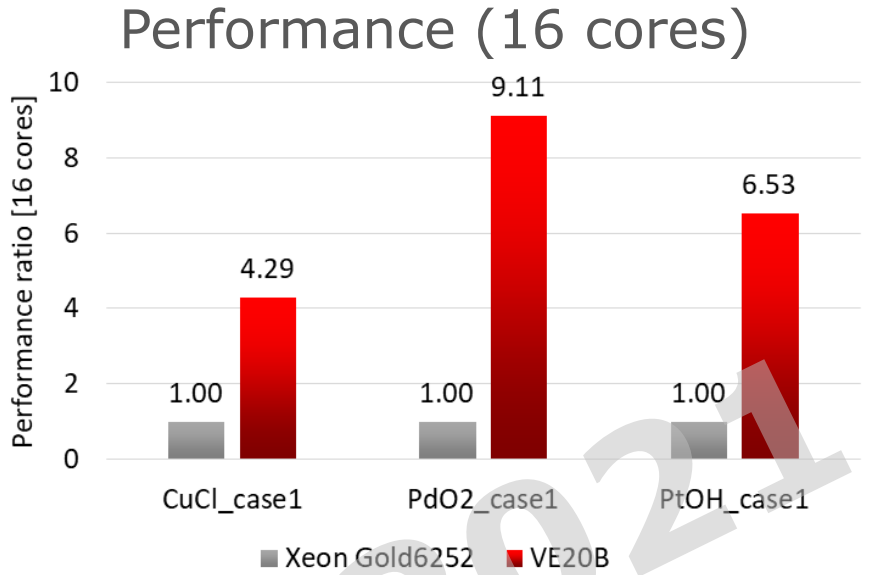


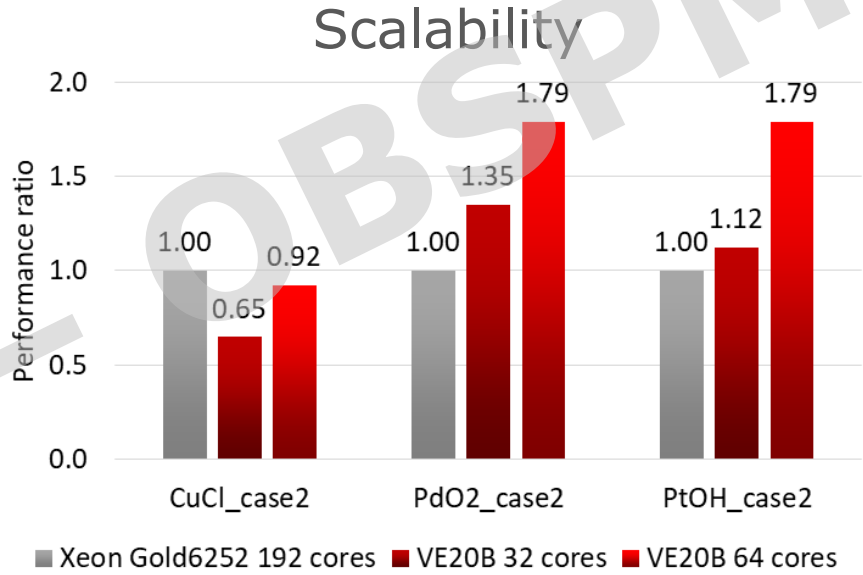


Performance

VASP 1/2

The Vienna Ab initio Simulation Package (VASP)
A copyright-protected software for atomic scale materials modelling, e.g. electronic structure calculations and quantum-mechanical molecular dynamics, from first principles. (reference: https://www.vasp.at/)

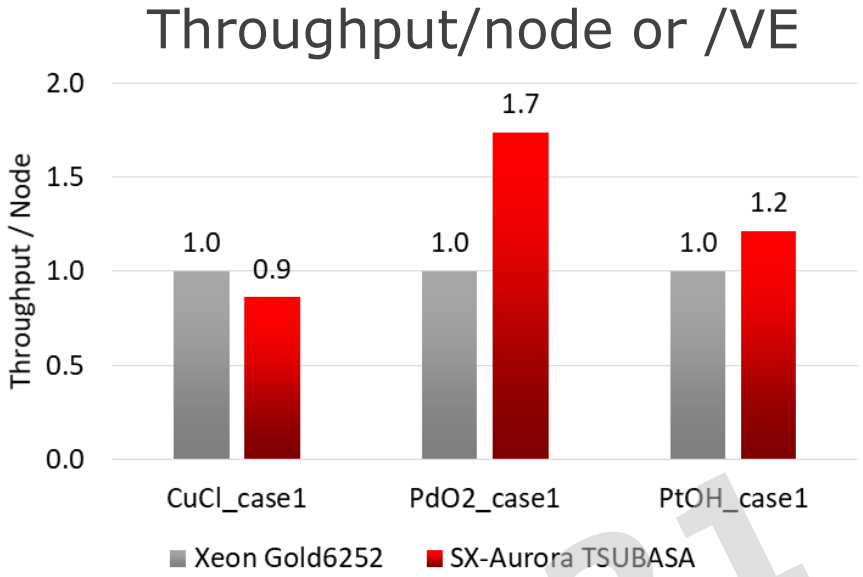


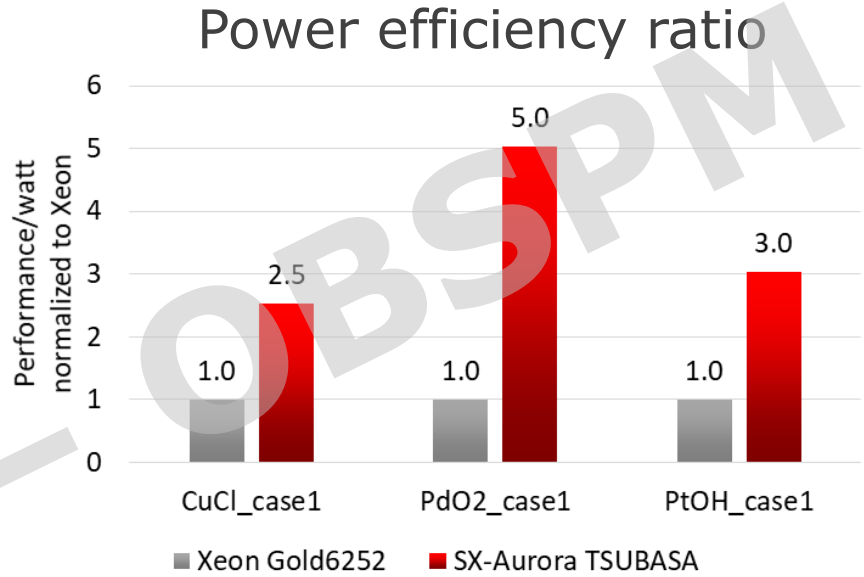


SX-Aurora TSUBASA
VASP: 5.4.4, patch 5.4.4.16052018
VE: Type 20B, Partitioning mode ON, NEC compiler 3.0.30, NEC MPI 2.6.0, NLC 2.0.0
VH: B401-8, VE20B x8, EPYC Rome x1, HDR200 x2, DLC/Air cooling
Intel Xeon Gold 6252
VASP: 5.4.4
Xeon: Gold 6252 (CLX Generation), 2 socket per node, 48 cores per node, 2.1GHz, air cooling

- Much higher (4.3~9.1x) VASP performance in the case of same used number of cores
- Much better VASP scalability provides higher sustained performance with fewer cores and nodes

VASP 2/2

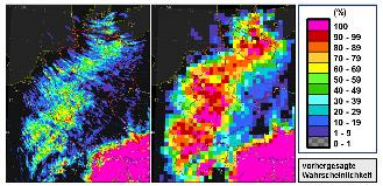




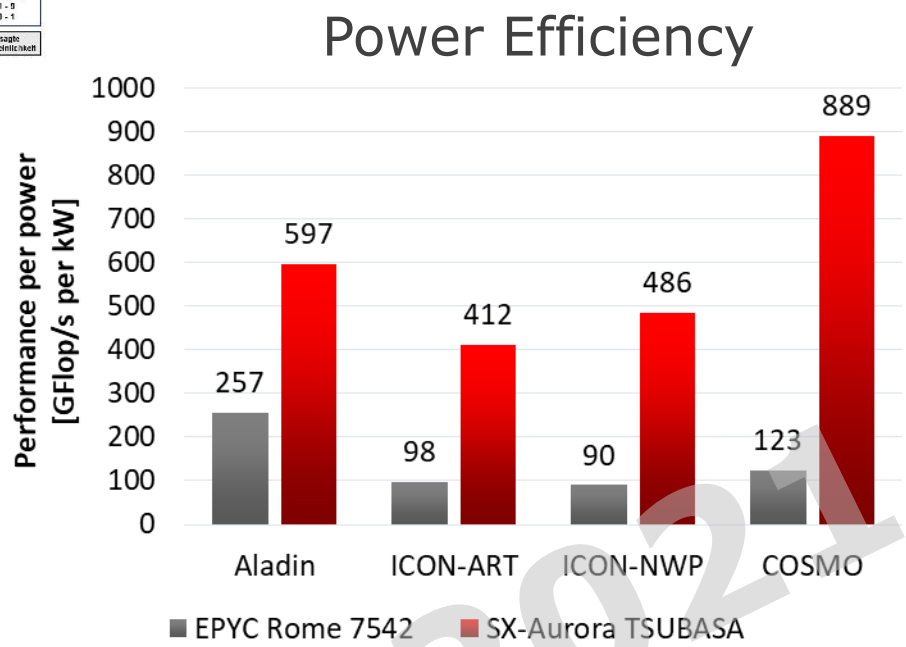
SX-Aurora TSUBASA
VASP: 5.4.4, patch 5.4.4.16052018
VE: Type 20B, Partitioning mode ON, NEC compiler 3.0.30, NEC MPI 2.6.0, NLC 2.0.0
VH: B401-8, VE20B x8, EPYC Rome x1, HDR200 x2, DLC/Air cooling
Intel Xeon Gold 6252
VASP: 5.4.4
Xeon: Gold 6252 (CLX Generation), 2 socket per node, 48 cores per node, 2.1GHz, air cooling

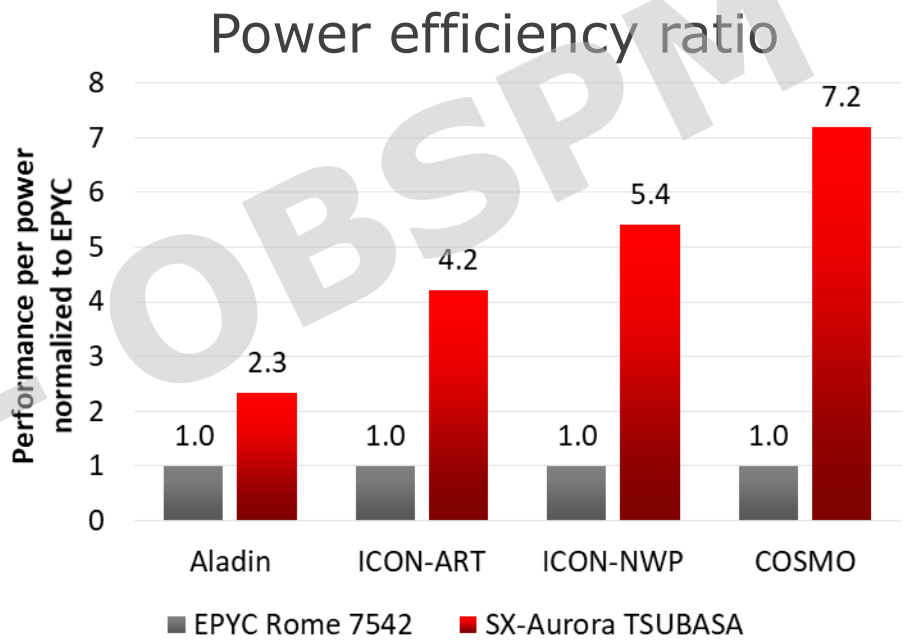
- Node throughput is same range or better than Xeon, but power is much lower (Xeon is around 500W per node, SX-Aurora TSUBASA is 170~200W per node
- Power efficiency is 2.5~5.0x better than Xeon

Weather forecast



dwd.de

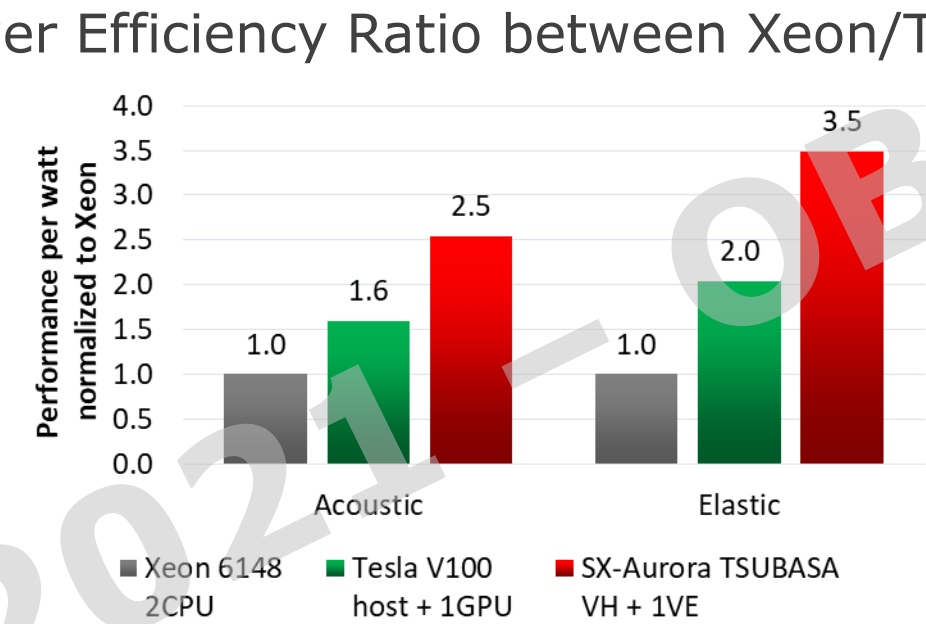




- **EPYC Rome 7542:** EPYC Rome 7542 32 cores/socket, 2.9GHz. 2 sockets per node
- **SX-Aurora TSUBASA:** VE10AE x8 / VH (single socket Rome)
- **ICON-ART:** Status as of 2019 for ICON-ART

- Power supply limitation is one of the big limiting factor of each system size
- Aurora contributes to accelerate meteorology codes within the power limitation
- For the major meteorology codes, Aurora provides 2-7x higher sustained performance with same power consumption

Resource Exploration



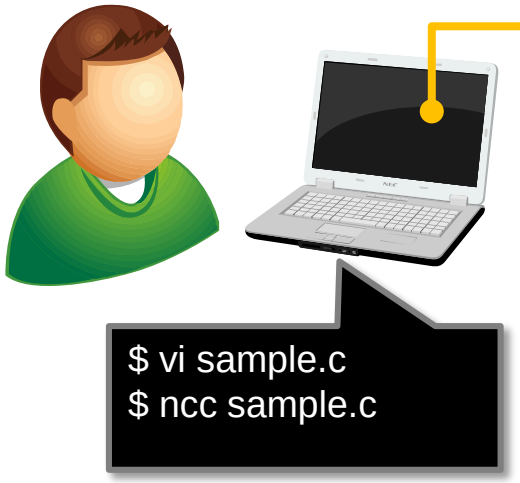
- **Xeon 6148:** Dual socket Xeon Gold 6148 (2 x 20 cores, 2.4GHz)
- **Tesla V100:** V100 x1 per host
- **SX-Aurora TSUBASA:** VE20B x1 per VH

- HPC capability is needed for finding oil, producing oil, and managing oil reservoirs
- Simulations of the resource exploration require higher memory bandwidth
- Aurora provides the highest power efficiency compared to X86, and GPGPU

Aurora programming environment

Ease of use

Programing Environment

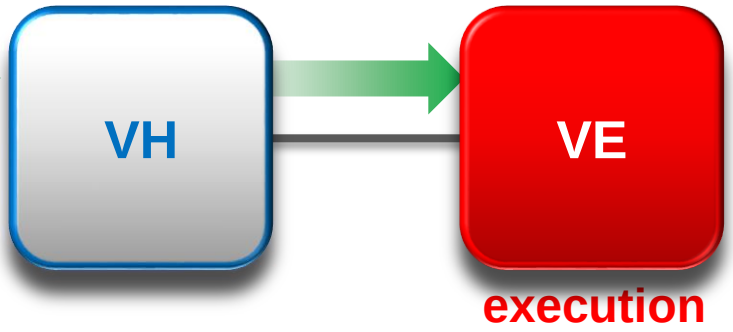


Vector Cross Compiler

automatic vectorization | automatic parallelization

Fortran:	F2003, F2008
C/C++:	C11/C++14/C++17
OpenMP:	OpenMP4.5
Library:	MPI 3.1, libc, BLAS, Lapack, etc
Debugger:	gdb, Eclipse PTP, ARM DDT
Tools:	PROGINF, FtraceViewer

Execution Environment



NEC Compiler for Vector Engine

Having powerful automatic vectorization and automatic parallelization

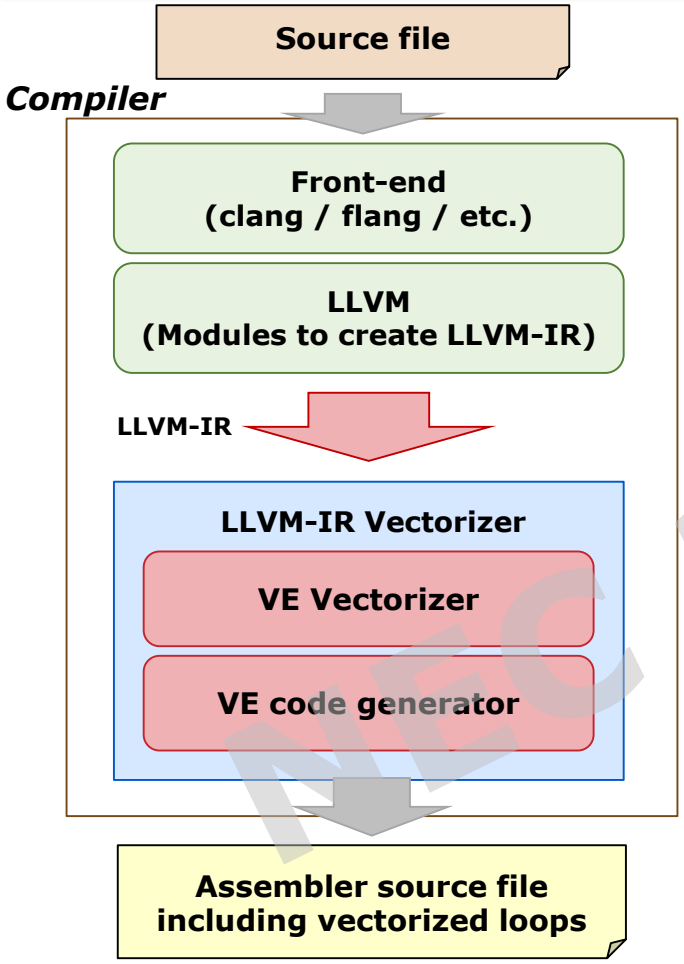
```
FORMAT LIST

13: P------>  for (j = 0; j < m; j++) {
14: |V----->    for (i = 0; i < n; i++) {
15: ||            if (f[i] > 0.0) {
16: ||              a[j][i] = b[j][i] + sin(c[j][i]);
17: ||            }
18: ||      G     sum += d[i] + e[idx[i]];
19: |V-----     }
20: P------   }

DIAGNOSTIC LIST

 LINE              DIAGNOSTIC MESSAGE

  13: par(1801): Parallel routine generated.: fun$1
  13: par(1803): Parallelized by "for".
  14: vec( 101): Vectorized loop.
  18: vec( 126): Idiom detected.: Sum
```

Outer-loop is parallelized and inner-loop is vectorized automatically.

- Variable-vector-length vectorization
- Masked vector operation can be vectorized
- Many vector mathematical functions are available
- Indirect accessed vector can be vectorized
- Reduction operation can be vectorized
- Many kinds of loop transformation to promote vectorization are available

LLVM-IR Vectorizer

Provide binary plugin-module and runtime-library to add automatic vectorization feature for VE into your compilers



LLVM-IR Vectorizer

- Soon to be released
- Includes vectorizer and code generator for VE
- Inputs LLVM-IR from memory or an IR-file and outputs an assembler source code for VE
- Applies automatic vectorization to LLVM-IR
- Has APIs to support compiler directives
- Runtime library including vector mathematical functions (sin,cos etc.) and vector matrix-multiply functions is bundled
- MLIR in the future

CUI Profiler suitable for VE architecture

PROGINF

```
     ********  Program  Information  ********
Real Time (sec)                :       121.233126
User Time (sec)                :       121.228955
Vector Time (sec)              :       106.934651
Inst. Count                    :    119280358861
V. Inst. Count                 :     29274454500
V. Element Count               :   6389370973939
V. Load Element Count          :   3141249840232
FLOP Count                     :   3182379290112
MOPS                           :     58637.969529
MOPS (Real)                    :     58635.323545
MFLOPS                         :     26251.407833
MFLOPS (Real)                  :     26250.223262
A. V. Length                   :       218.257559
V. Op. Ratio (%)               :        98.733828
L1 Cache Miss (sec)            :         0.639122
VLD LLC Hit Element Ratio (%)  :        73.051749
Memory Size Used (MB)          :     27532.000000

Start Time (date) : Sun Jul 22 20:09:51 2018 JST
End   Time (date) : Sun Jul 22 20:11:52 2018 JST
```

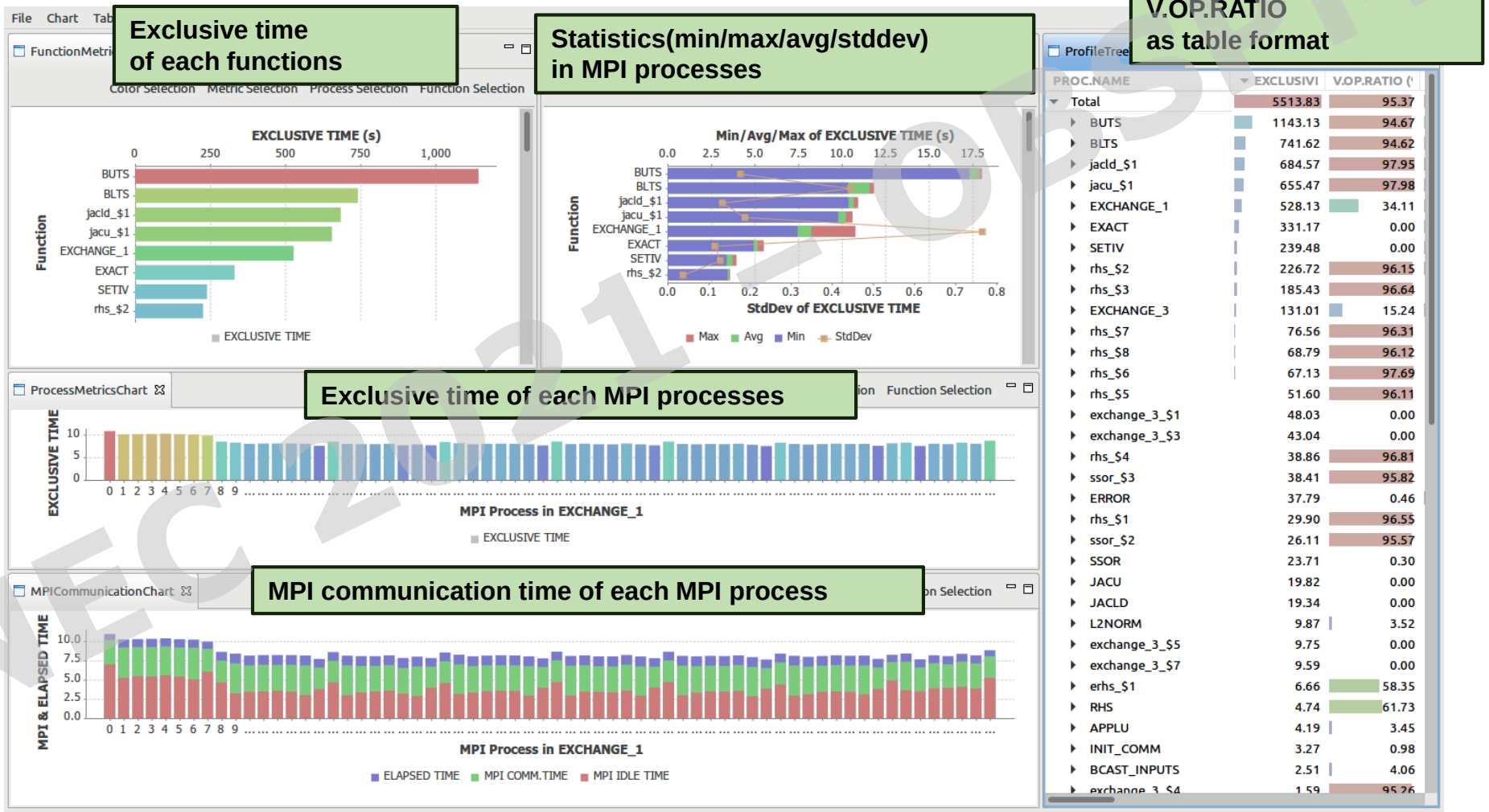
FTRACE

```
*----------------------*
  FTRACE ANALYSIS LIST
*----------------------*

Execution Date : Sun Jul 22 21:10:50 2018 JST
Total CPU Time : 0:02'02"434 (122.434 sec.)
```

FREQUENCY	EXCLUSIVE TIME[sec](%)	AVER.TIME [msec]	MOPS	MFLOPS	V.OP RATIO	AVER. V.LEN	VECTOR TIME	L1CACHE MISS	CPU PORT CONF	VLD LLC HIT E.%	PROC.NAME
512	58.110(47.5)	113.496	68380.5	30870.9	99.27	223.8	58.110	0.000	0.000	74.26	SUB1
510	31.763(25.9)	62.280	69474.6	31552.6	99.31	223.6	31.762	0.000	0.000	76.59	SUB2
2	9.056(7.4)	4527.963	3198.9	0.0	14.91	205.2	0.062	1.272	0.000	73.33	SUB3
459	7.732(6.3)	16.846	55793.0	23438.2	98.44	163.7	7.732	0.000	0.000	51.84	SUB4
459	7.037(5.7)	15.332	56239.0	26694.7	98.84	212.3	7.037	0.000	0.000	62.00	SUB5
2097152	6.667(5.4)	0.003	2816.0	322.1	22.88	256.0	0.282	0.283	0.000	100.00	SUB6
4	1.355(1.1)	338.818	19218.8	11094.9	98.75	243.9	1.355	0.000	0.000	0.82	SUB7
463	0.448(0.4)	0.968	57096.7	0.0	95.73	176.4	0.448	0.000	0.000	0.00	SUB8
1483	0.141(0.1)	0.095	18966.1	0.0	97.87	205.4	0.141	0.000	0.048	2.59	SUB9
2099270	0.122(0.1)	0.000	3056.8	17.2	0.00	0.0	0.000	0.000	0.000	0.00	SUB10
51	0.001(0.0)	0.017	667.2	0.0	0.00	0.0	0.000	0.001	0.000	0.00	SUB11
1	0.000(0.0)	0.292	444.7	0.6	0.01	8.0	0.000	0.000	0.000	0.00	MAIN_
4201150	122.434(100.0)	0.029	58071.6	25992.7	98.59	218.3	106.929	1.558	0.048	73.05	total

NEC SDK for VE - NEC Ftrace Viewer

GUI Profiler (Ftrace GUI-frontend)
OpenMP, MPI, and mix of them are supported.



NEC Numerical Library Collection (NLC)

NLC is a collection of mathematical libraries that powerfully supports the development of numerical simulation programs.

ASL Unified Interface

Fourier transforms and Random number generators

FFTW3 Interface

Interface library to use Fourier Transform functions of ASL with FFTW (version 3.x) API

ASL

Scientific library with a wide variety of algorithms for numerical/statistical calculations:
Linear algebra, Fourier transforms, Spline functions, Special functions, Approximation and interpolation, Numerical differentials and integration, Roots of equations, Basic statistics, etc.

Stencil Code Accelerator

Stencil Code Acceleration

BLAS / CBLAS

Basic linear algebra subprograms

LAPACK

Linear algebra package

ScaLAPACK

Scalable linear algebra package for distributed memory parallel programs

SBLAS

Sparse BLAS

HeteroSolver

Direct sparse solver

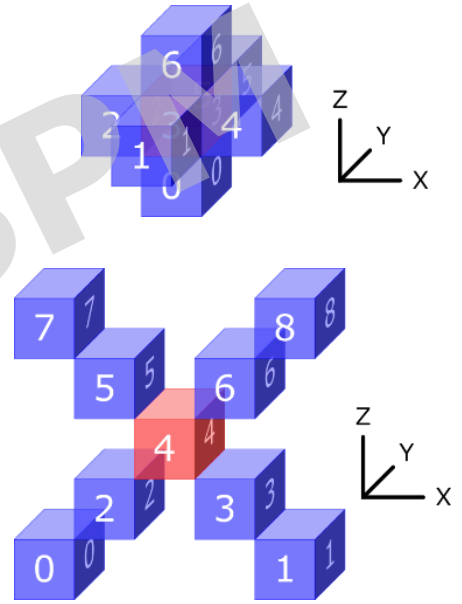
Stencil Code Accelerator (SCA) Library

What is "stencil code" ?

- A procedure pattern that frequently appears in scientific simulations, image processing, signal processing, deep learning, etc.
- Updates each element in a multidimensional array by referring to the neighbor elements.

$$B_{i,j,k} = cB_{i,j,k} + \sum_{r=-t}^{t} \sum_{q=-t}^{t} \sum_{p=-t}^{t} F_{p,q,r} A_{i+p,j+q,k+r}$$

➔ Requires significant performance of both computation and memory access.



[Stencil Shape Examples]

Domains Where Stencil Code Appears:

- Scientific Simulations: Fluid Dynamics, Thermal Analysis, Electromagnetic Field Analysis, Climate / Weather etc.
- Signal Processing: Audio, Sonar, Rader, Radio Telescopes, etc.
- Image / Volume Data Processing: Retouch, Data Compression, Recognition, Medical Diagnosis (Biopsy, CT, MRI, …), etc.
- Machine Learning: Deep Learning (Convolutional Neural Networks)

Stencil Code Accelerator

SCA is a library that highly accelerates stencil computations

Library Features

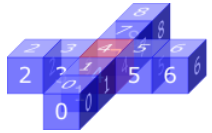
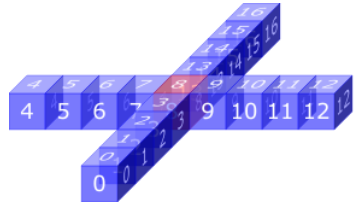
- SCA v1: Supports 56 stencil shapes (and stencil shapes can be composed)
- SCA v2: Arbitrary stencil shapes can be specified
- Supports 1, 2, and 3-dimensional data.
- Optimized for Vector Engine nearly to the limit.
- For C/C++ and Fortran.

Supported Stencil Shapes

SCA v1 provides widely usable 56 stencil shapes.

Stencil Shapes

- SCA supports the following types of stencil shapes:
  - {X,Y,Z}-Directional 
  - {XY,XZ,YZ}-Planer 
  - {XY,XZ,YZ}-Axial 
  - {XY,XZ,YZ}-Diagonal 
  - XYZ-Volumetric 
  - XYZ-Axial 

- SCA supports the following sizes for each type:
  - 1 
  - 2 
  - 3 
  - 4 

SCA Example

Example of Stencil Code – using SCA

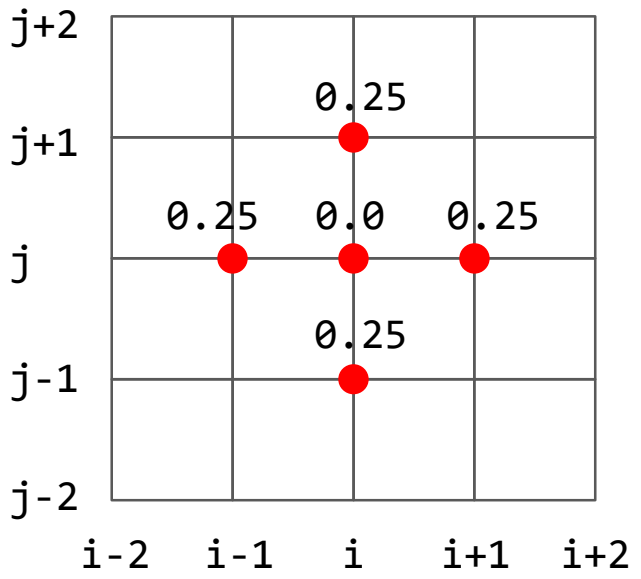
```
real :: a(0:ni+1,0:nj+1)
real :: b(1:ni,1:nj)
real,parameter :: c(5)=(/ &
   0.25,0.25,0.0,0.25,0.25/)
    :
do itr=1,maxitr
    :
   call sca_compute_with_1x1ya_s( &
      ni,nj,1, &
      ni+2,nj+2,a(1,1), &
      ni,nj,b(1,1),c,0.0)
    :
   a(:,:)=b(:,:)
end do
```

Laplace Equation

$$\frac{\partial}{\partial x^2}A + \frac{\partial}{\partial y^2}A = 0$$

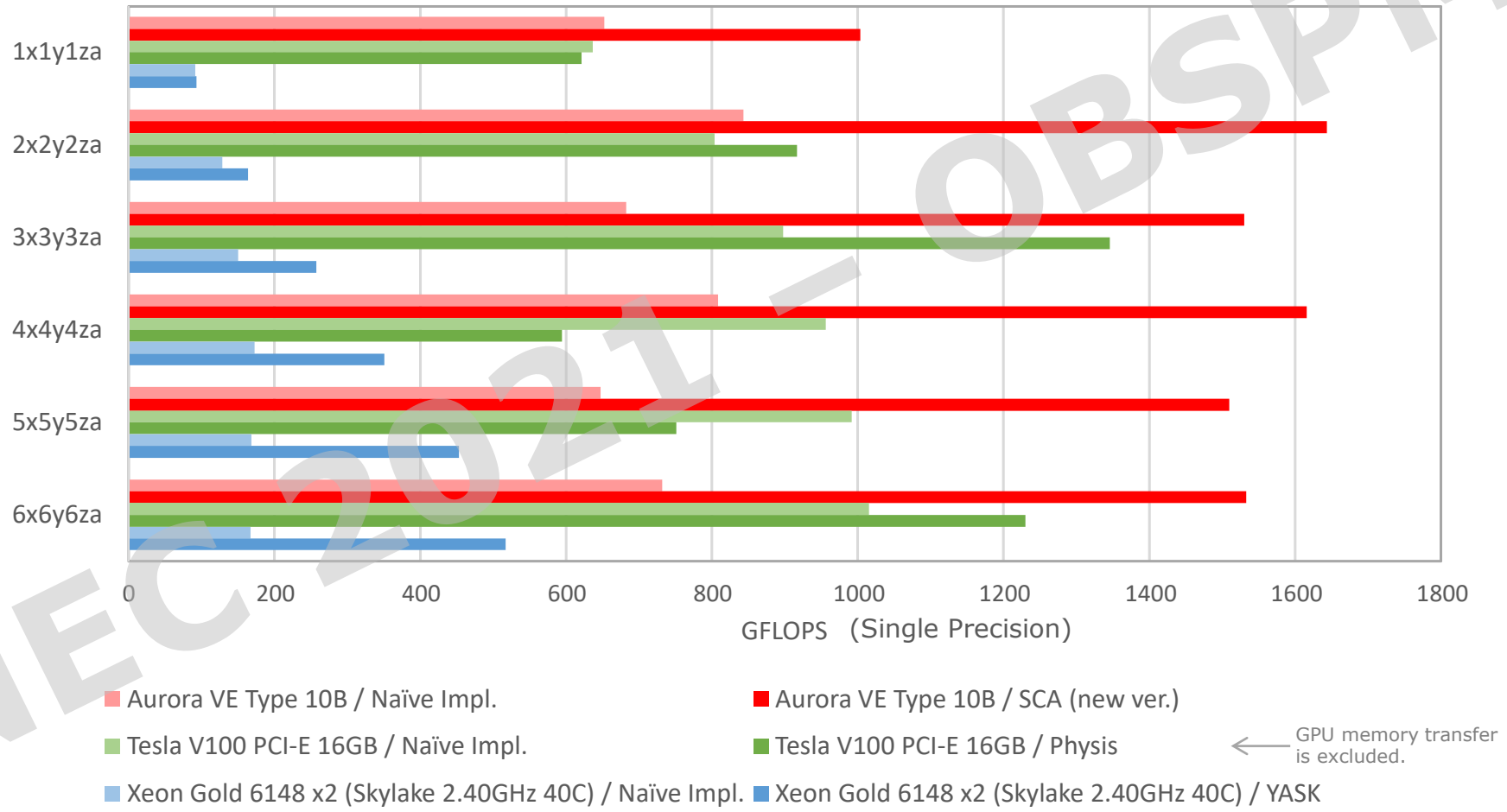
discretization
finite-difference

$$b_{i,j} = \frac{1}{4}\left(a_{i,j-1} + a_{i-1,j} + a_{i+1,j} + a_{i,j+1}\right)$$



SCA Performance Comparison

SCA on Vector Engine shows the highest performance.



NEC SX-Aurora SW environment in a nutshell

Compiler

- ncc, nc++, nfort
  - Autovectorizing, inner loop, highly optimizing
- clang, clang++ : free, open source LLVM
  - Intrinsics, Vectorization, OMP Target

Parallelism

- Vectorization
- OpenMP
- MPI + Hybrid MPI

Libraries

- NLC NEC Numeric Library Collection
  - ASL incl. FFTW3
  - BLAS, LAPACK, ScaLAPACK, BLACS
  - SBLAS sparse BLAS
  - Heterosolver: direct sparse solver
  - Stencil Code Accelerator Library

Offloading

- VH-call
- VEO / AVEO
- VEDA
  - CUDA-like, with long-vector kernels
- OpenMP Target in LLVM
  - VH→VE
  - VE→VH

Python

- NLCpy → NUMPY on VE
- SOL → PyTorch
- Py-veo

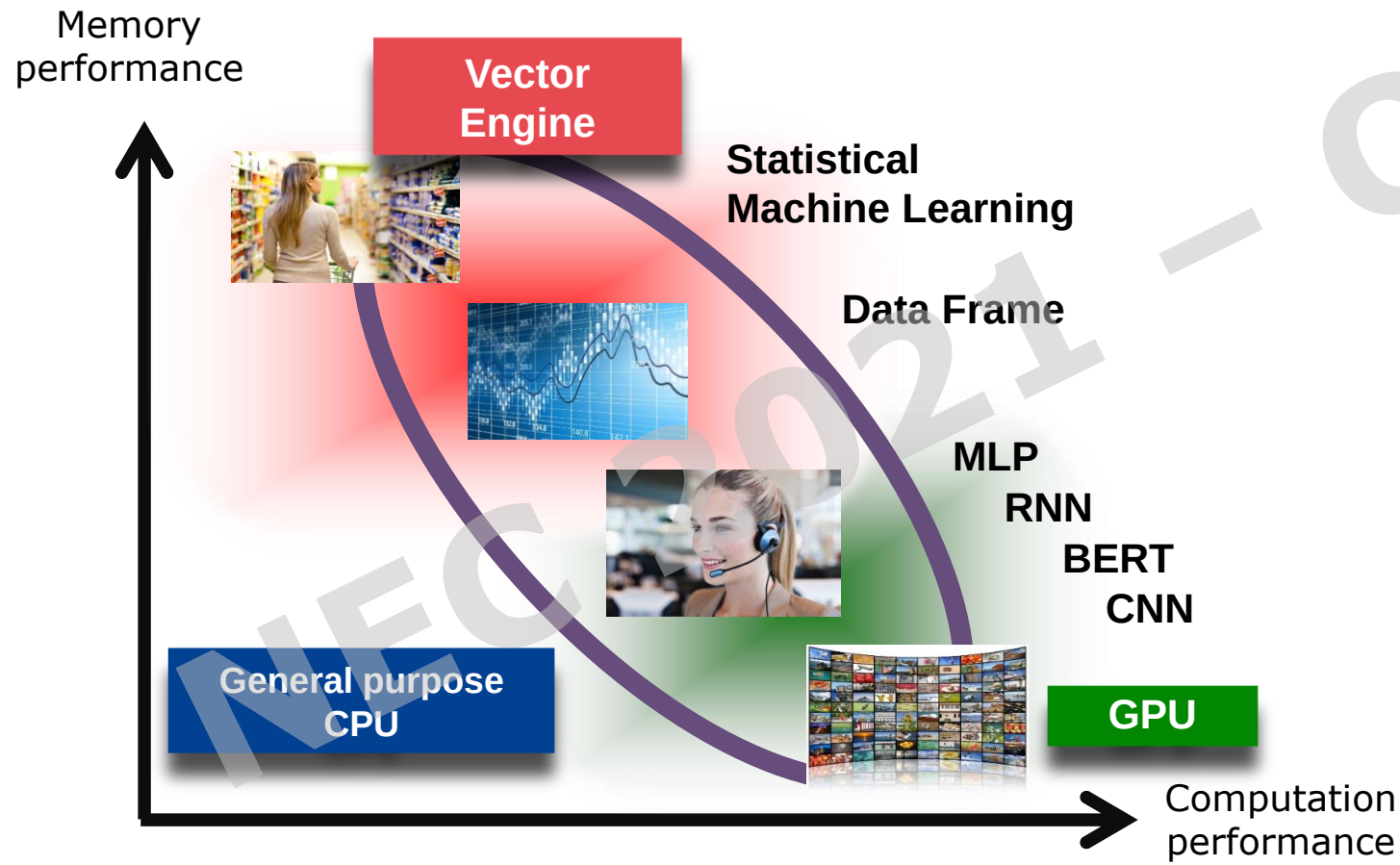
Tools

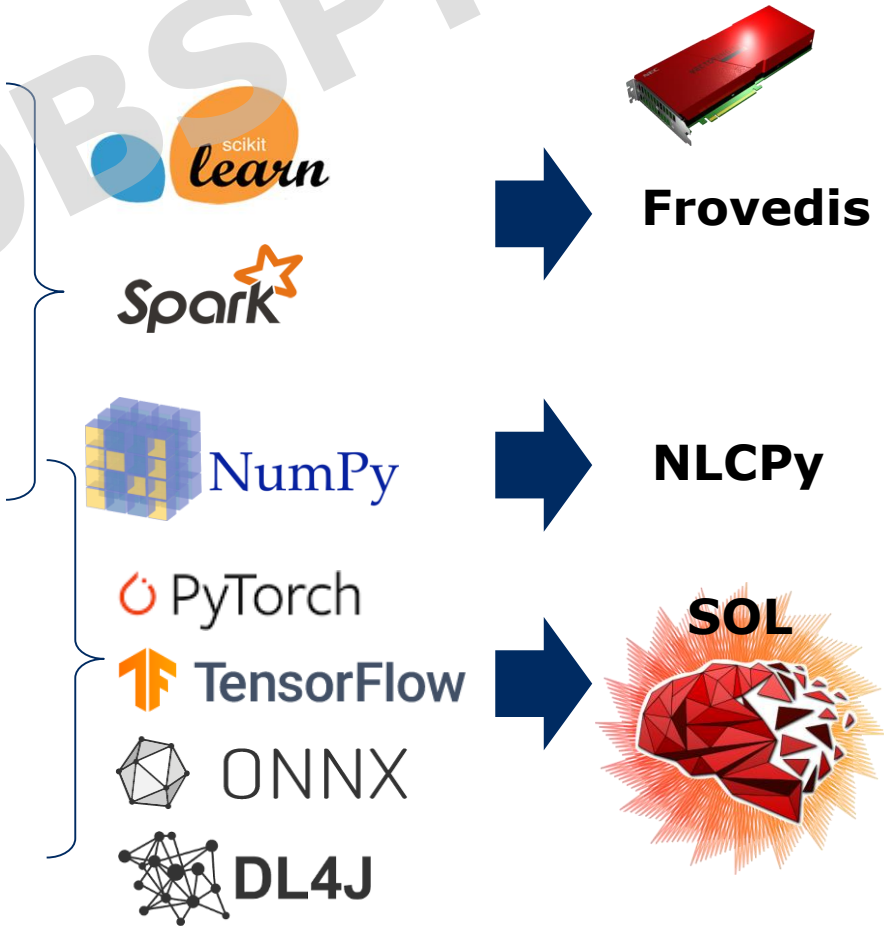
- TAU, PAPI
- Vftrace
- DDT

ML & DL

AI/ML on SX-Aurora TSUBASA

- AI/ML that requires memory performance can be well accelerated
- NEC provides frameworks for easy utilization





NLCPy

NLCPy is a package for accelerating the performance of NumPy scripts.
Enables NumPy scripts to compute on Vector Engine or SX-Aurora TSUBASA.
Provides subset of NumPy's API.

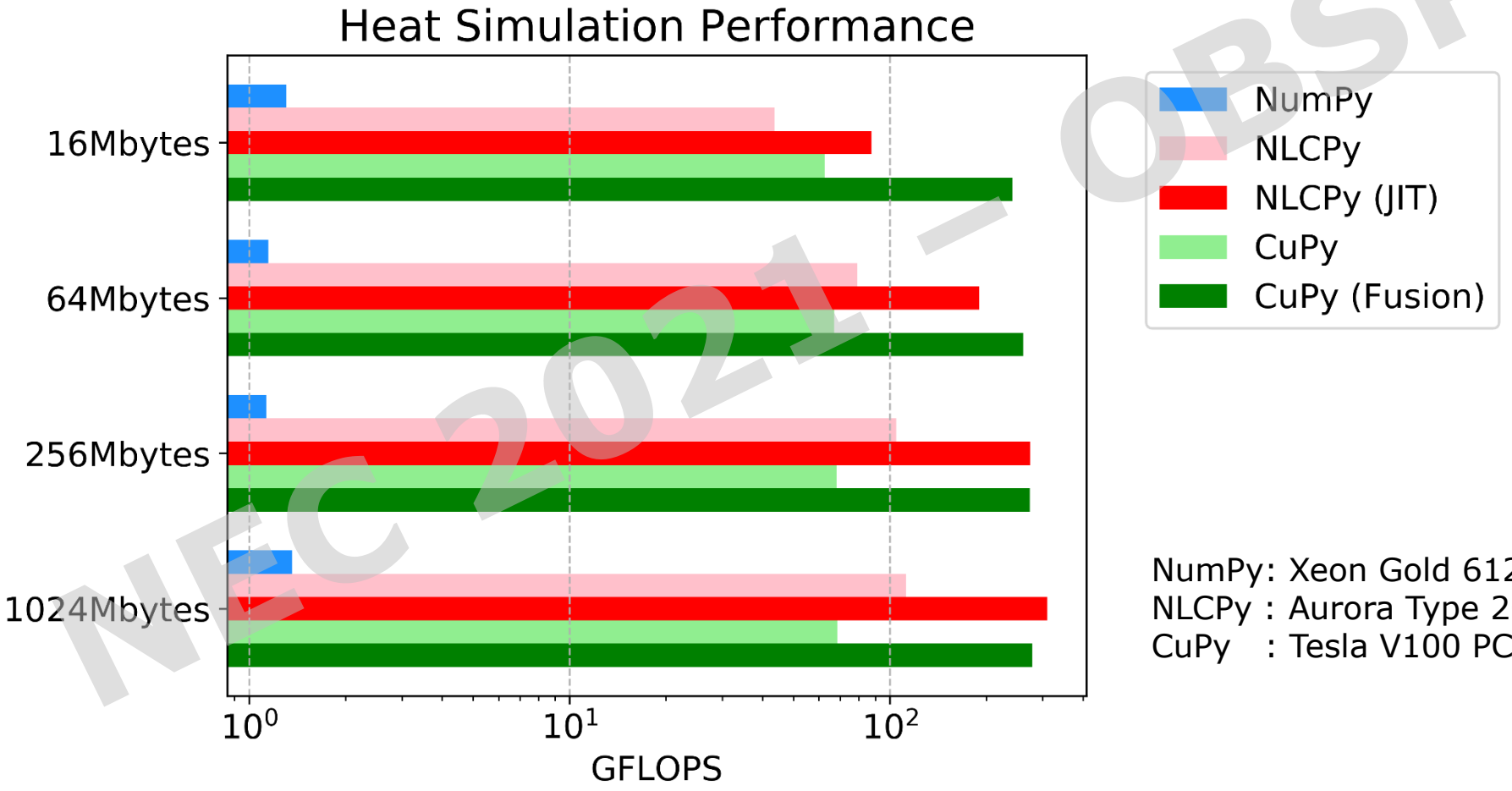
Example of Python Script

```
import numpy as np

a = np.rand.rand(1000, 1000)
b = np.rand.rand(1000, 1000)

c = a + b
d = np.dot(a, b)
```

```
import nlcpy as np

a = np.rand.rand(1000, 1000)
b = np.rand.rand(1000, 1000)

c = a + b
d = np.dot(a, b)
```

Benchmark Result (2-D Heat Simulation)

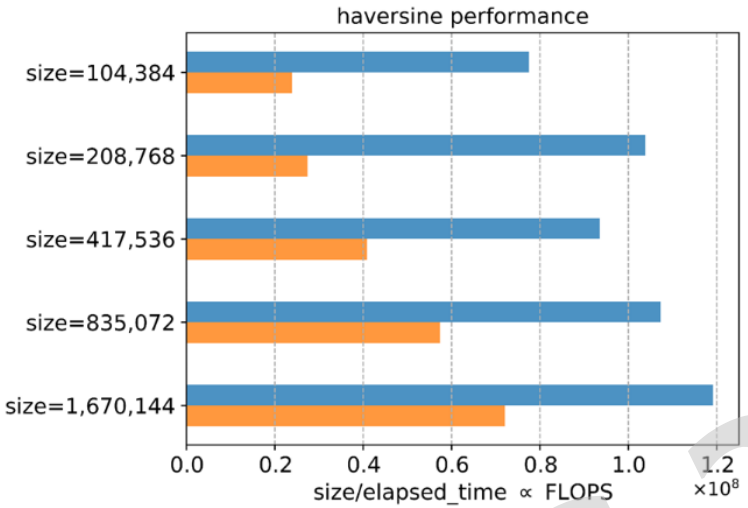
NLCPy shows higher performance than NumPy(CPU)
When array size is 1024Mbytes, NLCPy shows slightly higher performance than CuPy(GPU)



NumPy: Xeon Gold 6126, 48 cores
NLCPy : Aurora Type 20B, 8 cores
CuPy : Tesla V100 PCI-E, 16GB

NLCPy vs NumPy

NLCPy delivers 3.8x higher Performance compared to NumPy for Haversine
NLCPy delivers 37x higher Performance compared to NumPy for 2-D Heat Equation

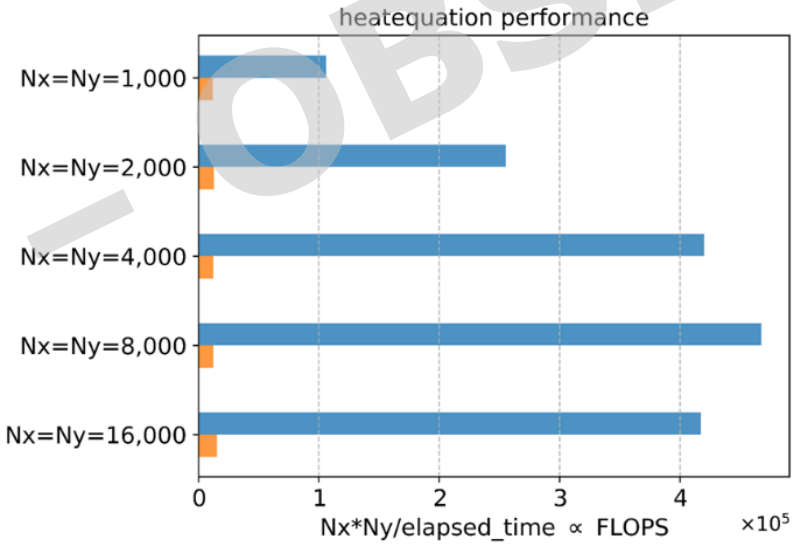


size: Number of sets of latitudes and longitudes

Processors:
- NumPy
Xeon Gold 6126 x 2 sockets
(Skylake 2.6 GHz, 48 cores)

- NLCPy
Aurora 1.4GHz, 8 cores

faster



Nx: Number of grid points in x-direction
Ny: Number of grid points in y-direction

Processors:
- NumPy
Xeon Gold 6126 x 2 sockets
(Skylake 2.6 GHz, 48 cores)

- NLCPy
Aurora 1.4GHz, 8 cores

faster

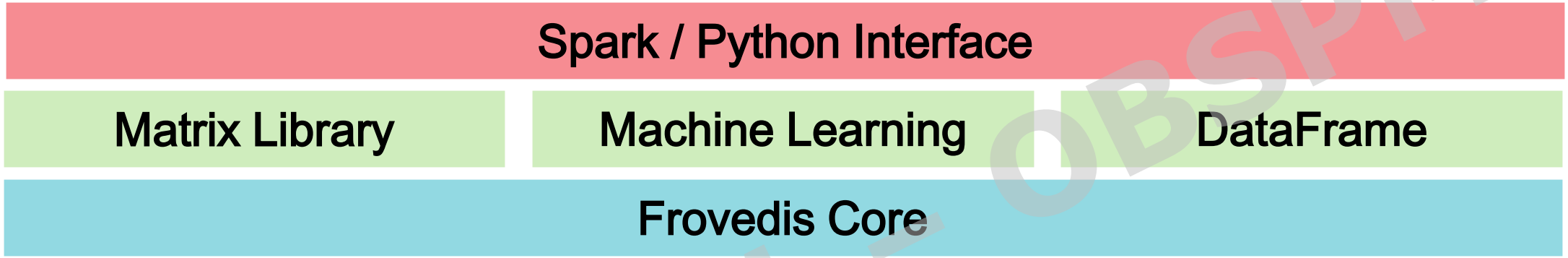
The benchmarking programs are based on the following site:

https://github.com/weld-project/split-annotations/tree/master/python/benchmarks/haversine

https://benchpress.readthedocs.io/autodoc_benchmarks/heat_equation.html#heat-equation-python-numpy

Frovedis: FRamework Of VEctorized and DIStributed data analytics

- C++ framework similar to Apache Spark



Open Source!

github.com/frovedis

- Supports Spark/Python (scikit-learn) interface on SX-Aurora TSUBASA (also works on X86)

Frovedis for ML

Normal Spark Code

```
…
import org.apache.spark.mllib.classification.LogisticRegressionWithSGD
…
val model = LogisticRegressionWithSGD.train(data)
…
```

Frovedis Code

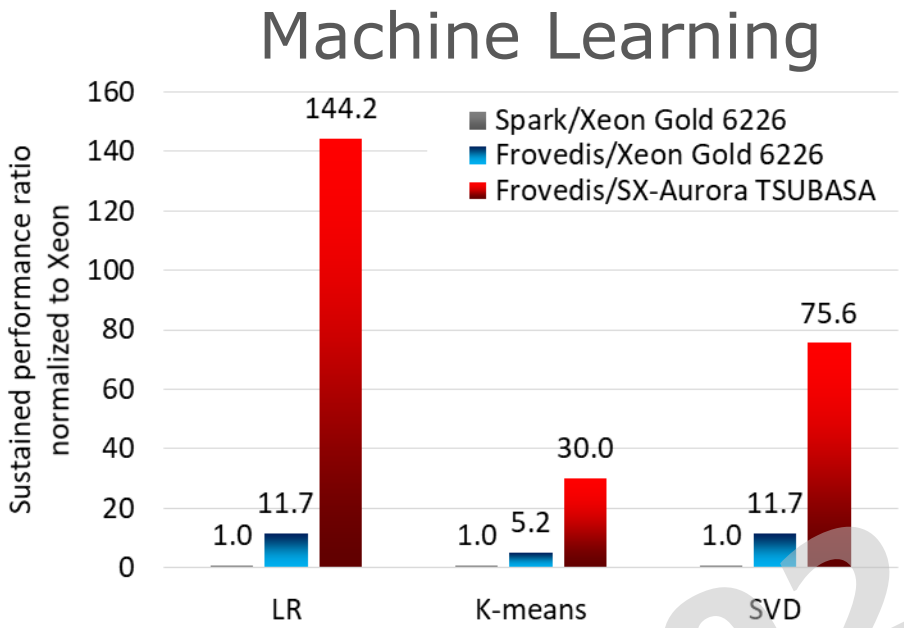
```
…
import com.nec.frovedis.mllib.classificaiton.LogisticRegressionWithSGD
…
FrovedisServer.initialize(…)
Val model = LogisticRegressionWithSGD.train(data)
FrovedisServer.shut_down()
…
```

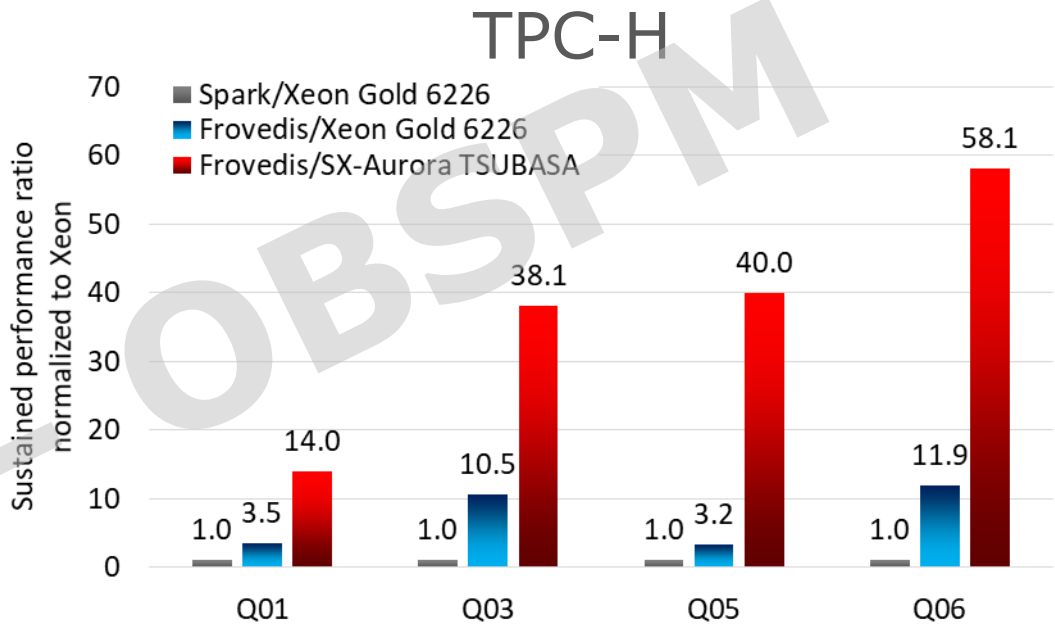
Change import

Start server

Stop server

ML Acceleration





- **Xeon Gold 6226:** 12 cores/socket, 2 socket/node, CentOS8.1
- **SX-Aurora TSUBASA:** A311-8, VE Type 10BE, compiler 3.0.7, MPI 2.9.0
- **Spark:** 3.0.0
- **Frovedis:** as of 2020/09/08

- Frovedis accelerates Spark executions on the Xeon processor (up to 12x)
- Frovedis on SX-Aurora TSUBASA provides much higher sustained performance compared to Xeon (up to 144x)

Machine Learning Library

Implemented with Frovedis Core and Matrix Library

- ✓ Supports both dense and sparse data
- ✓ Sparse data support is important in large scale machine learning

Supported algorithms:

- Linear model
  - Logistic Regression
  - Multinominal Logistic Regression
  - Linear Regression
  - Linear SVM
- ALS
- K-means
- Preprocessing
  - SVD, PCA
- Word2vec
- Factorization Machines
- Decision Tree
- Naïve Bayes
- Graph algorithms
  - Shortest Path, PageRank, Connected Components

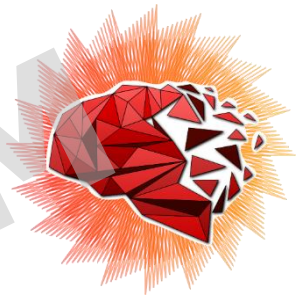
Under development:

- Frequent Pattern Mining
- Spectral Clustering
- Hierarchical Clustering
- Latent Dirichlet Allocation
- Deep Learning (MLP, CNN)
- Random Forest
- Gradient Boosting Decision Tree

We will support more!

Deep Learning with PyTorch and Tensorflow

SOL is an independent easy to plug NEC library for Deep Learning on SX-Aurora TSUBASA



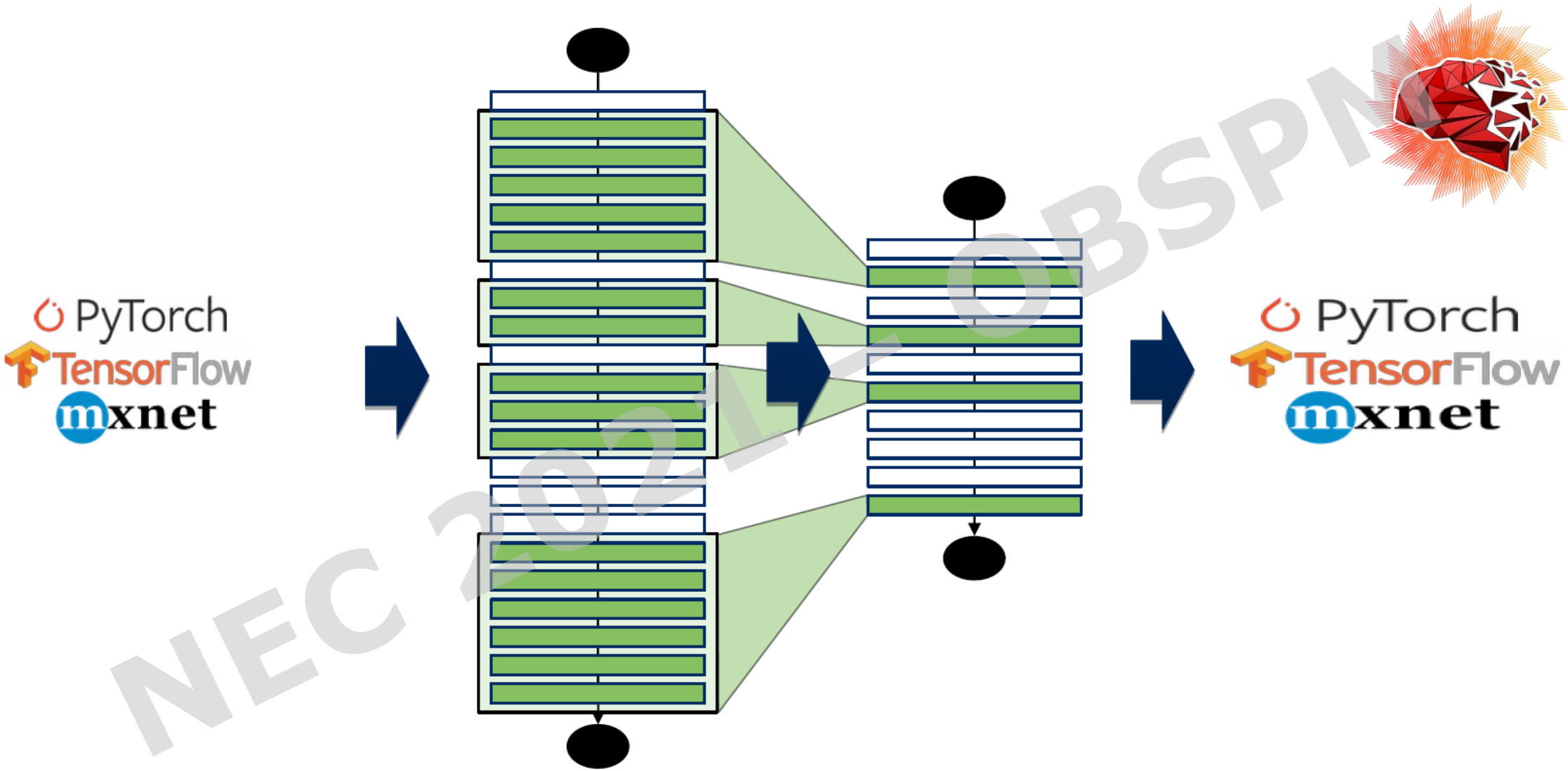
- **Optimizes NN and generates** optimized code for hardware backend
- **Few lines required for porting** to use DL frameworks
- **HPC/AI use case : can generate a C/C++ library for inference that can be used in your main application simulation.**
  Case where batch size is equal to one, more memory bound than computationally bound.

Currently supported framework :

VE-TensorFlow :

- Open source project for native support of Tensorflow 2 on SX-Aurora TSUBASA.
- https://github.com/sx-aurora-dev/tensorflow

SOL – Transparent NN Optimizer

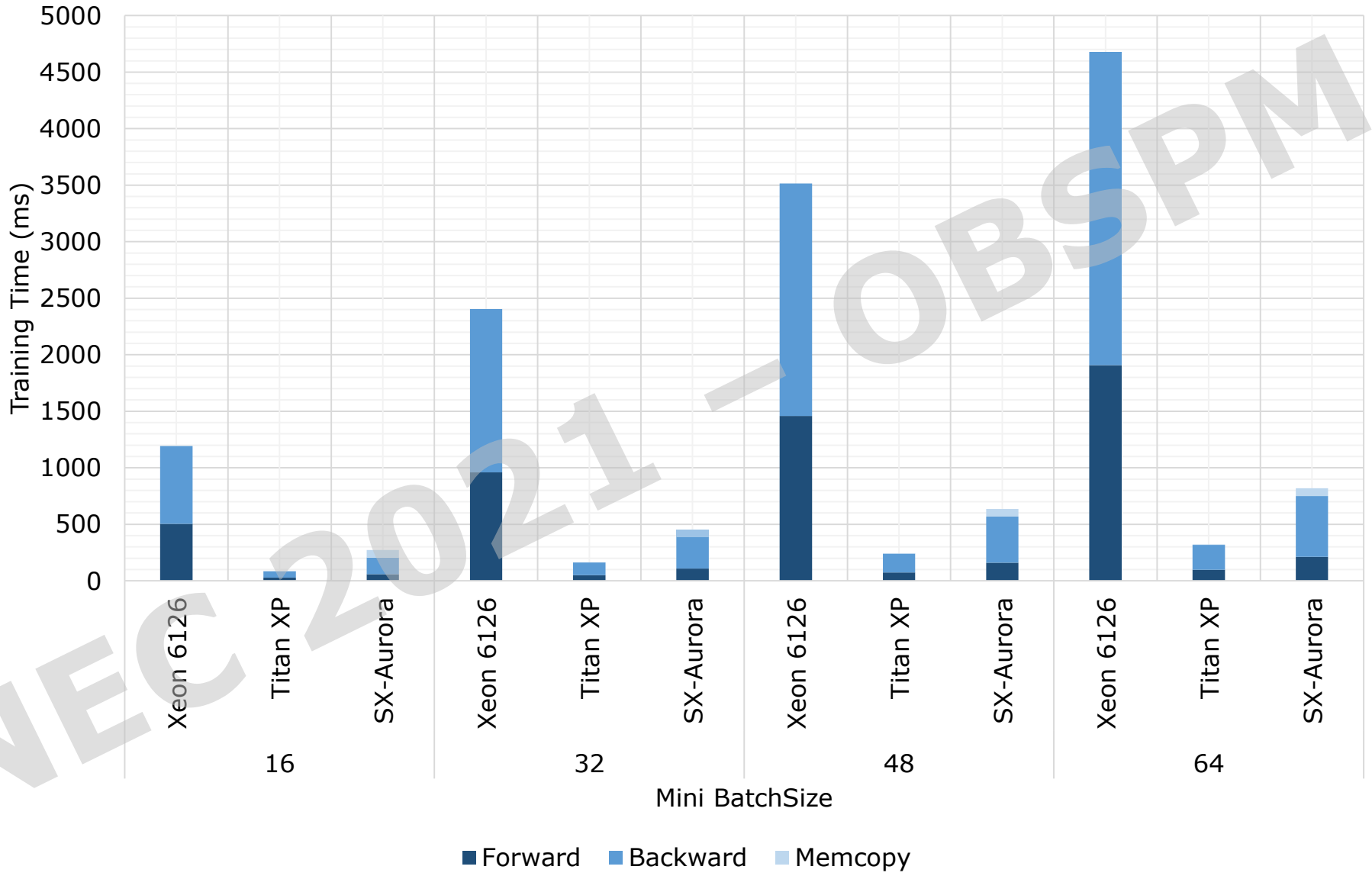


Deep Learning with pytorch and Tensorflow : SOL

Very easy to use :


```
import torch
from torchvision import models
import sol.pytorch as sol

py_model   = models.__dict__["…"]()
input      = torch.rand(0, 32, 224, 224)
sol_model = sol.optimize(py_model, input.size(),batchsize=1)
sol_model.load_state_dict(py_model.state_dict())
sol.device.set(sol.device.ve, 0)
output = sol_model(input)
```

Training on DenseNet 121 with SOL on SX-Aurora TSUBASA



ML/DL summary

- SX-Aurora TSUBASA is an HPC accelerator which supports ML and DL workloads easily with NLCPy, Frovedis, SOL and VE-TensorFlow.
- Frovedis shows huge performance improvements on several machine learning use cases.
- HPC/AI convergence/acceleration will be available with TensorFlow support


Orchestrating a brighter world
NEC