NEC Vector Supercomputer

SX-Aurora TSUBASA Generation 2

October 2020

Future Technology for Today

Supercomputing has a long history, yet it continues to evolve at a rapid rate. Not only applications but also computer-architectures have become more complex. We increasingly cooperate with scientists to overcome such obstacles, to bridge the gap between computer science and computational science, which leads to the application itself.

The NEC SX-Aurora TSUBASA is an integral part of our strategy. SX-Aurora TSUBASA's goal is to provide both high efficiency and power efficiency whilst still being simple to use, and it is based on standard programming paradigms and very elaborate software components like compilers and libraries.

NEC SX-Aurora TSUBASA is the legitimate successor of our former very successful SX-product line. It is the only remaining real vector architecture, but at the same time it is a different concept in terms of openness and availability. While the former SX-systems were heavy and expensive, the "Aurora" is available in the form of a PCIe-based cards, thus ensuring compliance with modern hardware interface standards. While the former SX-systems featured their own operating system, the "Aurora" appears to the user like a normal Linux system, and it surely is one. Moreover, the system interoperates openly with all other kinds of systems, so it can easily be integrated into every environment.

NEC is honoured to support the German Weather Service (DWD) and the Czech Weather Service (CHMI) as customers. Both are using our vector architecture. Our customers value our solutions due to the superior power efficiency of our systems. They continuously run mission-critical operations, which are consequently very demanding. And from their feedback and requirements we ourselves learn a lot to improve the product and our long-term HPC-strategy.

In recent years new aspects of supercomputing became very important, with those applications emerging dubbed as "Deep Learning" or "Machine Learning" or even "Artificial Intelligence". While we have to think carefully about the societal and even ethical aspects of these developments, we also see the huge potential for unprecedented achievements in science and technology. Especially in times like this, one would specifically think about those numerous possible applications in biology, pharmacology and medicine.

The computational challenges of such applications are often similar to other more traditional fields of HPC, which we know well from close to 40 years of experience in the market. Our technology is perfectly optimized and has always been, and we prove this in the frameworks we provide like Tensorflow or PyTorch, which excel in performance on "Aurora". NEC's research labs all over the world are actively engaged in continuously improve our developments.

We are innovating both based on our experience, but also with an eye on the ever-changing technology landscape. We also utilize this technology as a contribution to a better society. We summarize this approach in our business message: Orchestrating a brighter world.

Akio Ikeda
Deputy General Manager
AI Platform Division

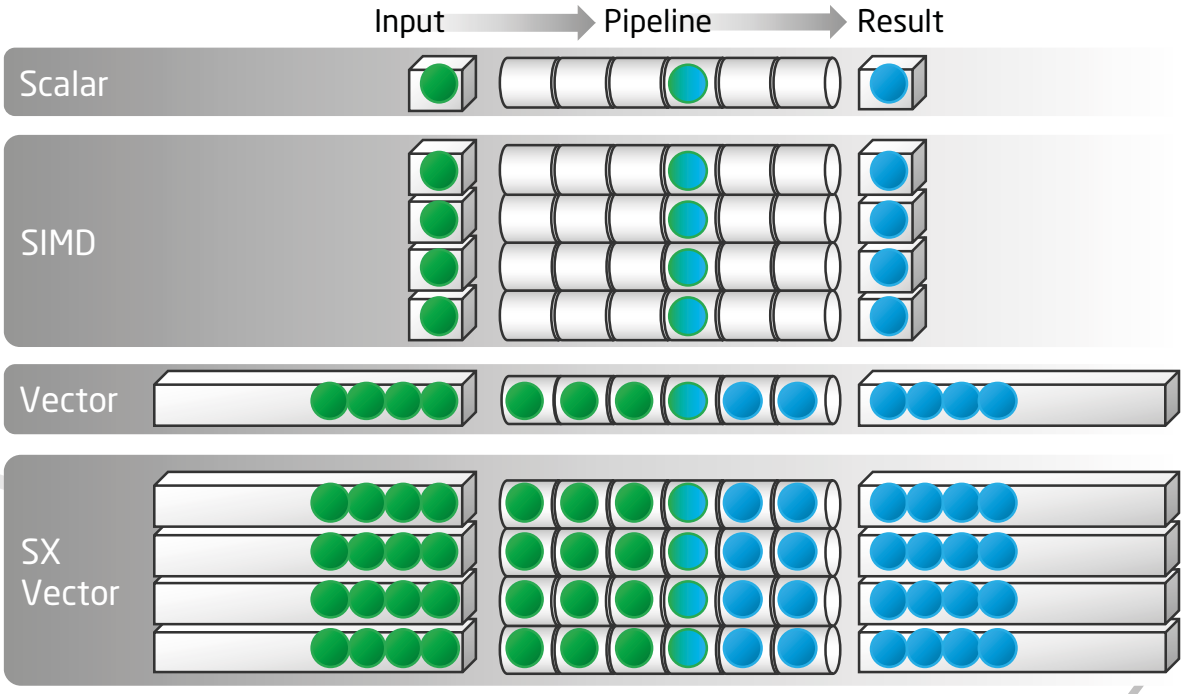
Vector Computing: A New Paradigm, With a Long History

HPC and AI have always been at the cutting edge of technology looking for an advantage.

Now is the time for a breakthrough. NEC's legacy in vector computing has always been an engine and a story of success, and our vector technology addresses issues found in other systems and provides this breakthrough in HPC.

Technical or scientific problems are expressed in the language of mathematics, and this mathematics needs to be reflected in the way the applications for solving these problems are written. Algorithms need to be optimized, and the variables need to be organized in memory accordingly. The vector computing paradigm is simple: "I have to execute a mathematical operation. On which elements, grids, variables, particles, equations, structures, can I apply it simultaneously?".

If this is your way of thinking, you will inevitably write vector code. For example, it is one of the best practices in C to organize variables as a structure of arrays, not as an array of structures. It is also this understanding that leads to the idea of "domain specific languages" (DSL) – frameworks for describing actions to be applied to a whole field. NEC provides the hardware and software environment necessary to apply this vector paradigm to the solution of real-world problems.



Arithmetic operations of each architecture

NEC SX-Aurora TSUBASA – The New Generation Vector Architecture

The SX Vector Architecture from Past to Present

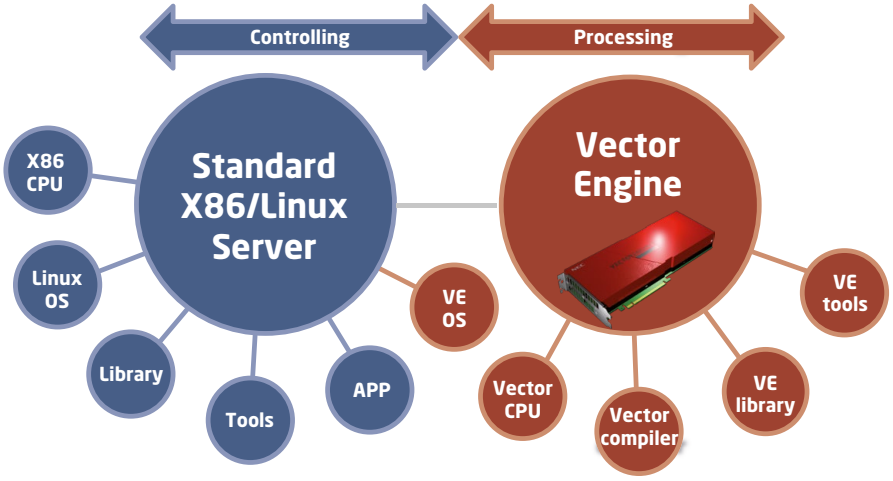
NEC has developed and provided the SX supercomputer series since 1983. The development concept of "pursuing higher sustained performance with higher usability" has not been changed from our very first machines, SX-1 and SX-2, launched more than 35 years ago. Each generation of the SX supercomputer used cutting edge technology then available to implement the SX vector architecture into the processors of the then current generation. With the latest generation, the SX vector supercomputer adopted a new hybrid architecture, the SX-Aurora TSUBASA architecture. This hybrid architecture consists of a computation part and an OS function part. The heart of the new SX architecture is the vector engine (VE) contained in the vector host (VH). The VE executes complete applications while the VH mainly provides OS functions for connected VEs.

Series of the SX vector Supercomptuer

Model	 SX-1/2	 SX-3	 SX-4	 SX-5	 SX-6	 SX-7	 SX-8/8R	 SX-9	 SX-ACE	 SX-Aurora TSUBASA	 SX-Aurora TSUBASA Gen2
Year	1983	1989	1994	1998	2001	2002	2004	2007	2013	2018	2020
Technology	Bipolar	Bipolar	350 nm	250 nm	150 nm	150 nm	90 nm	65 nm	28 nm	16 nm	16 nm
CPU frequency	166 MHz	340 MHz	125 MHz	250 MHz	500 MHz	552 MHz	1.0 GHz	3.2 GHz	1.0 GHz	~1.6 GHz	~1.6 GHz
CPU performance	1.3 GF	5.5 GF	2.0 GF	8.0 GF	8.0 GF	8.8 GF	16.0 GF	102.4 GF	256.0 GF	~2.45 TF	~3.07 TF
CPU Memory bandwidth	10.7 GB/s	12.8 GB/s	16.0 GB/s	64.0 GB/s	32.0 GB/s	35.3 GB/s	64.0 GB/s	256.0 GB/s	256.0 GB/s	~1.22TB/s	~1.53TB/s

SX-Aurora TSUBASA Architecture

The heart of the NEC SX Aurora TSUBASA architecture – the vector engine (VE) – offers a hitherto unparalleled performance especially for memory-bound applications. As a standard PCIe card, it fits effortlessly in a standard x86 server host environment. Each vector CPU has access to six extremely high-bandwidth HBM2 memory modules, which allows for unprecedented performance for memory-intensive applications. With the first generation of SX-Aurora TSUBASA, the world's first implementation of a CPU design with six HBM2 memory modules using a "chip-on-wafer-on-sub-strate" technology (CoWoS) applications were provided with the world-record memory bandwidth of 1.22TB/s. The second generation of SX-Aurora TSUBASA equipped with the second generation VE is continuously leading the world's highest memory bandwidth of 1.53TB/s.



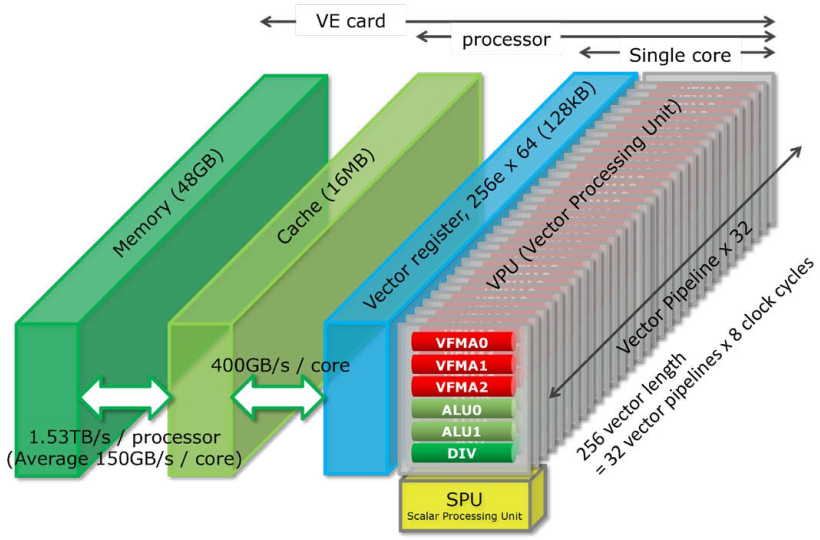
SX-Aurora TSUBASA architecture

- Vector Processor
- 1.53TB/s Memory Bandwidth
- Low Power 200W/VE

The second generation VE, Vector Engine Type 20 (VE20) is equipped to the latest model of the SX-Aurora TSUBASA supercomputer. The VE20 generation is mainly improved memory bandwidth of 1.53TB/s, and the number of cores per processor. NEC continuously pursues to provide higher sustained performance for the memory bandwidth bound applications by continuously providing highest-level memory bandwidth per processer in each VE generations.

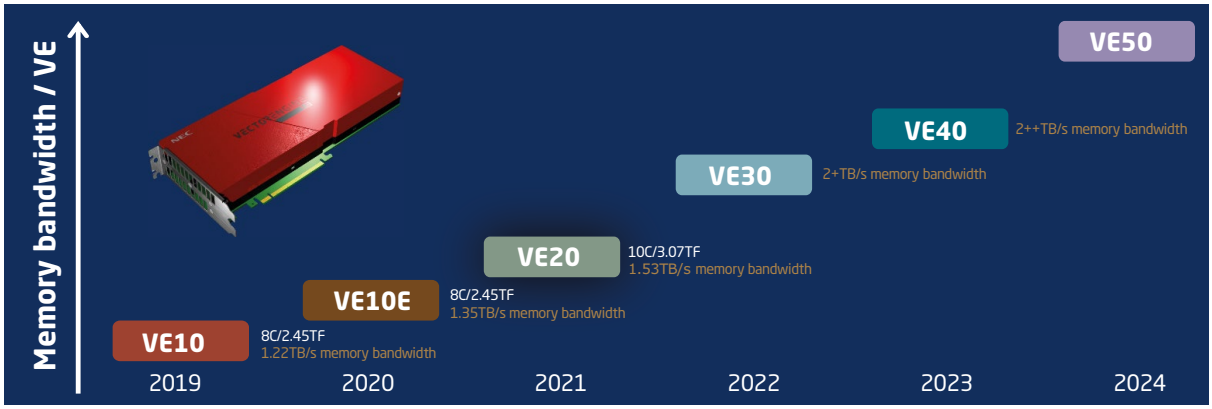
The VE20 generation has two VE types, Vector Engine Type 20A and Type 20B. VE Type 20A is 3.07TF peak performance with 10 vector cores, and VE Type 20B is 2.45TF peak performance with eight vector cores. Each vector core and 16MB shared cache are connected by a two dimensional mesh network and its bandwidth per vector core is 400GB/s maximum. The vector processor is based upon a 16 nm FinFET process technology for extremely high performance and low power consumption.

The vector core on the VE is the most powerful single core in HPC available today, thus keeping the design philosophy from the previous SX series. It achieves the industry-leading performance of 307GF per core, and an average memory bandwidth of 150 GB/s per core for the 10 cores configuration per the VE processor.
A vector engine processor has up to 10 independent vector cores. Each vector core mainly consists of 32 vector pipelines, and three FAM units ("fused multiply-add") are implemented into each vector pipeline. The vector lengths of 256 is processed by 32 vector pipelines with eight clock cycles. 64 fully functional vector registers per core – with 256 entries of 8 bytes width each – can feed the functional units with data or receive results from those, thus being able to handle double-precision data at full speed.



Vector Engine processor: single core and memory subsystem

The next generation, VE30 is planned to be released as the successor of the VE20 generation. The main improvement from the predecessor is memory bandwidth of 2+TB/s, and the memory subsystem of the processor will be reinforced to achieve higher sustained performance with lower power consumption.



Roadmap of Vector Engine

SX-Aurora TSUBASA Product Portfolio

Vector Engine

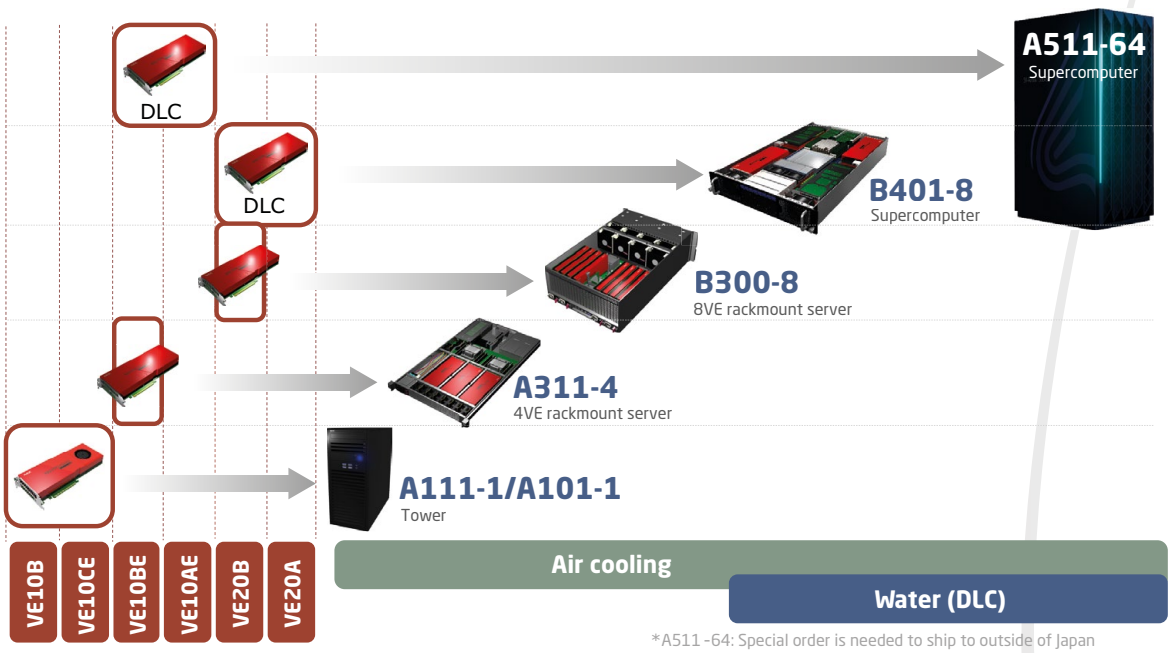
The latest vector engine models from the second generation are VE Type 20A and 20B. The VE Type 20A processor has 10 cores running at 1.6GHz with the 1.53TB/s high memory bandwidth. Since each core performance is more than 300GFlops, the VE processor reaches 3.07TFlops peak performance. VE Type 20B has less number of cores, thus allowing for a higher maximum memory bandwidth per core for memory bandwidth intensive applications.

Memory Capacity	Memory Bandwidth			
	1.53TB/s		20B	20A
48GB	1.35TB/s	10BE*	10AE*	*10AE is 1.584GHz *10BE is 1.408GHz
	1.22TB/s	10B		
24GB	1.00TB/s	10CE		
		2.15TF	2.45TF	3.07TF
	Frequency	1.4GHz	1.6GHz	
	Cores	8core		10core

Vector Engine specifications

SX-Aurora TSUBASA

Currently, SX-Aurora TSUBASA consists of three product types, which are the 500/400 series for large-scale supercomputers, the 300 series for standard operation environment, and the 100 series as the entry or development model.



*A511-64: Special order is needed to ship to outside of Japan

SX-Aurora TSUBASA product portfolio

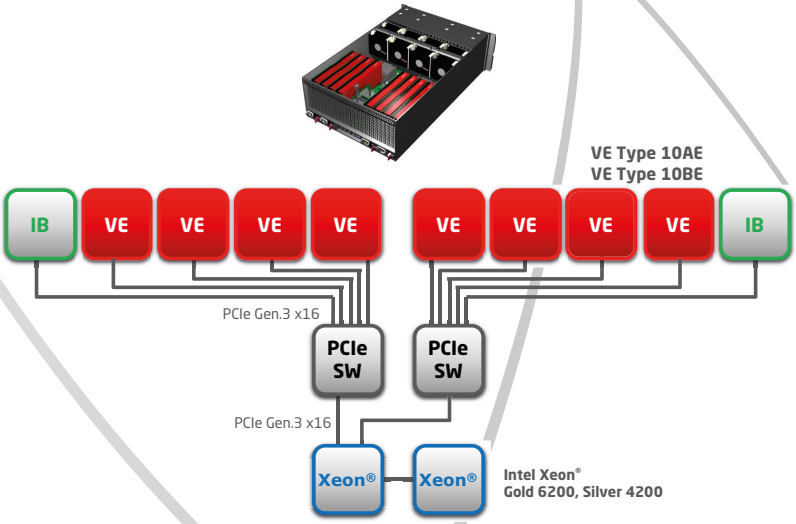
The latest product of the SX-Aurora TSUBASA 500 and 400 series are A511-64 and B401-8. To fulfil current requirements of large-scale supercomputers, direct-liquid-cooling (DLC) is applied to those models. The SX-Aurora TSUBASA 300 series is the mid-range line, and the latest models are B300-8 and A311-4. These models can operate in a standard rack with standard air-cooling environment. The SX-Aurora TSUBASA 100 series consists of the A111-1 and A101-1 models, providing VE capability to be used in a desktop environment.

SX-Aurora TSUBASA A511-64 Supercomputer

At the high-end of our range is SX-Aurora TSUBASA A511-64. This model is single rack product offering an optimized full-rack cooling fit for large-scale supercomputer solutions. The A511-64 mainly consists of eight VHs per rack of 4U height each and with eight VE cards with DLC. This model supports VE Type 10AE and Type 10BE. In order to reduce the cooling costs, an inlet water temperature of up to 40 °C is supported (hot water cooling).

The figure below shows one A511-64 vector host configuration with eight VE cards, two Xeon processors, and two Infiniband HCAs. Two PCIe switches connect each VE, Xeon, and IB HCA. Each VE card is equipped with a DLC cold plate, while other components including the Xeon processors are cooled by air.

SX-Aurora TSUBASA A511-64

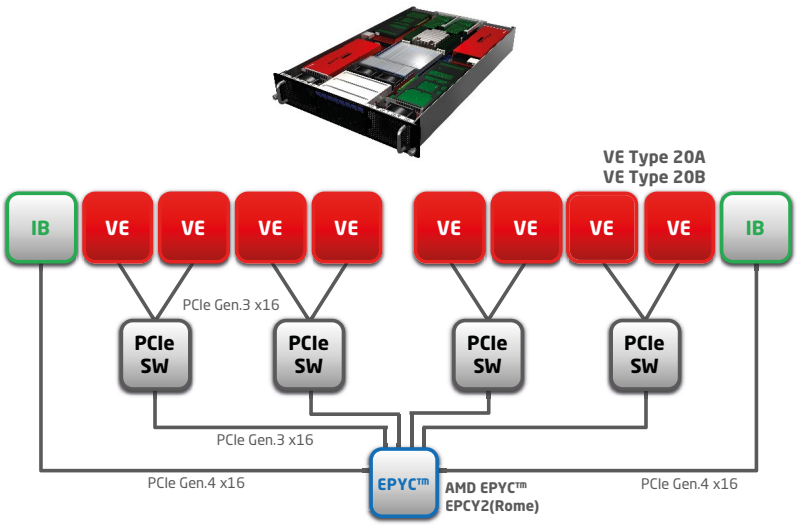


SX-Aurora TSUBASA A511-64: topology of single VH

SX-Aurora TSUBASA B401-8 Supercomputer

The B401-8 model provides both high vector processing capability and higher density. This model equipped with eight VE cards per VH. The main difference between the A511-64 and B300-8 models having the same eight VEs per VH is the higher density of 2U height per chassis, the AMD EPYC host processor, and a rack mount 2U server implementation. The below figure shows a block diagram of the B401-8 model. Each pair of VE cards is connected to PCIe switch. The supported VE cards are Type 20A and Type 20B.

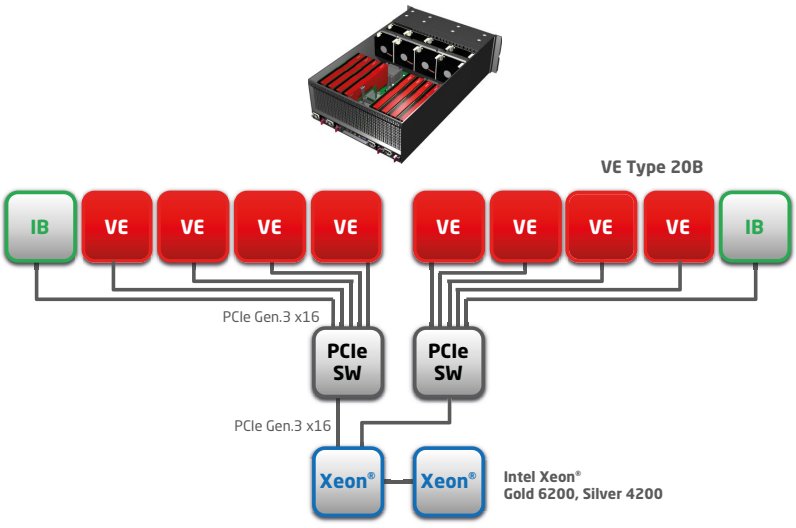
In the B401-8, DLC is applied to eight VE cards and the AMD EPYC processor, while all other components are cooled by airflow. Up to 40°C water and air can be used for the cooling. Thus the NEC design is a hybrid cooling concept that optimizes the ever-existing trade-off between implementation cost, floor load, cooling cost, maintainability. For the DLC cooling, NEC manifolds support up to 18 VH units per rack. In total, up to 144 VE cards can be mounted per rack and in the case of using VE Type 20A, peak performance reaches 442TFlops with more than 220TB/s aggregated memory bandwidth.



SX-Aurora TSUBASA B401-8: topology of single VH

SX-Aurora TSUBASA B300-8 Rackmount Server

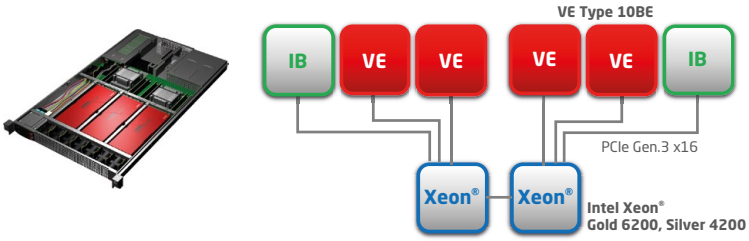
The SX-Aurora TSUBASA B300-8 model is designed for an air-cooled large-scale SX-Aurora TSUBASA system. Its configuration is up to eight VE cards in an 4U height VH, and the VH configuration is basically the same as the A511-64 DLC supercomputer, supporting VE Type 20B. As the following figure shows, eight VE cards and two Infiniband HCAs are connected to Xeon processors via two PCIe switches. This model allows up to 40°C airflow temperature, thus enabling to use an indirect water cooling with up to 35°C water temperature in combination with side coolers. This contributes to reducing power consumption of chillers for the cooling.



SX-Aurora TSUBASA B300-8: topology of single VH

SX-Aurora TSUBASA A311-4 Rackmount Server

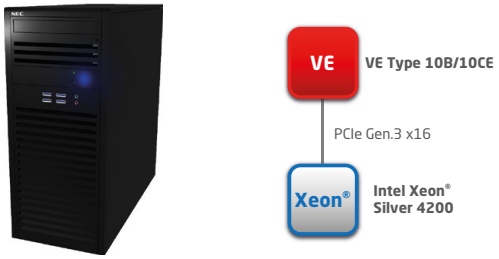
A block diagram of A311-4 is shown in the following figure. The A311-4 model consists of four VE cards of type 10BE and two Xeon processors. Moreover, up to two IB HCAs can be deployed with this server to enhance interconnect bandwidth. All components are implemented into 1U form factor.



SX-Aurora TSUBASA A311-4: topology of single VH

SX-Aurora TSUBASA A111-1/A101-1 Tower

The SX-Aurora TSUBASA A111-1/A101-1 models are entry models of SX-Aurora TSUBASA, specially adapted for desktop computing and software development. It mainly consists of one Xeon processor and one VE card of VE type 10CE and 10B.



SX-Aurora TSUBASA A111-1/A101-1: topology of single VH

Vector Programming with Accomplished Development Tools

A supercomputer is a tool to increase the productivity of researchers and developers. Writing software should be as simple as possible, and the efficiency and the effectiveness of the researcher is most important. NEC will always use and promote the use of standard programming languages and parallelization paradigms and not require exotic proprietary paradigms like CUDA.

For the NEC SX-Aurora TSUBASA software developers will find a plethora of accomplished development tools and an elaborate programming environment for writing superb vector code. Software development is conducted on the vector host, a standard LINUX-system familiar to every user. A vector cross-compiler translates the code into an executable SX-Aurora TSUBASA binary. At runtime, the VE operating system (VEOS) loads the binary into the vector engine, where it then executes.

[Compiler diagnostic message]

```
nfort -mparallel -O3 -report-all -c ex.f90
nfort: par(1801): ex.f90, line 6: Parallel routine generated.: DEL$1
nfort: par(1803): ex.f90, line 6: Parallelized by "do".
nfort: opt(1592): ex.f90, line 7: Outer loop unrolled inside inner loop.: J
nfort: vec( 101): ex.f90, line 8: Vectorized loop.
```

[Format list]

```
 1:           subroutine del( l, m, n, f, del_f, a )
 2:
 3:           real(kind=8) :: f(l,m,n), del_f(l,m,n)
 4:           real(kind=8) :: a(7)
 5:
 6: P------>  do k = 2, n-1
 7: |U----->    do j = 2, m-1
 8: ||V---->      do i = 2, l-1
 9: |||      F       del_f(i,j,k) = &
10: |||                a(1) * f(i,j,k-1) + a(2) * f(i,j-1,k)                    &
11: |||              + a(3) * f(i-1,j,k) + a(4) * f(i,j  ,k) + a(5) * f(i+1,j,k) &
12: |||                                  + a(6) * f(i,j+1,k) + a(7) * f(i,j,k+1)
13: ||V----        end do
14: |U-----      end do
15: P------    end do
16:
17:           return
18:           end
```

Automatic vectorization by NEC compiler

The vector cross-compiler supports advanced automatic vectorization and parallelization for industry-leading sustained performance and highly optimized MPI libraries in a GNU/Linux environment. The NEC SX-Aurora TSUBASA programming environment comprises a feature-rich compiler supporting several modern programming language standards, as well as a set of precompiled mathematical libraries for easy development of scientific code, including BLAS, LAPACK, ScaLAPACK, and FFTW3 Interface.

SX-Aurora TSUBASA programming environment

C	C11 (ISO/IEC 9899:2011)
C++	C++14 (ISO/IEC 14882:2014)
Fortran	Fortran 2003 (ISO/IEC 1539-1:2004) Fortran 2008 (ISO/IEC 1539-1:2010)
OpenMP	Version 4.5
C Library	GNU C Library
MPI	Version 3.1
Numeric libraries	BLAS, LAPACK, ScaLAPACK, FFTW3 Interface
Further tools	GNU profiler (gprof) GNU debugger (gdb) ECLIPSE Parallel Tools Platform (PTP) Ftrace Viewer / PROGINF

Superior Energy-Efficiency of the Vector Architecture

In recent years, power consumption has become a major topic in the HPC-market. The new generations of semiconductor technology lead to increasing core-counts, but at the same time increasing power consumption. This cannot be sustained in the long run, both for cost-reasons and for the environment's sake. Consequently, architectures need to be evaluated with regard to their energy efficiency.

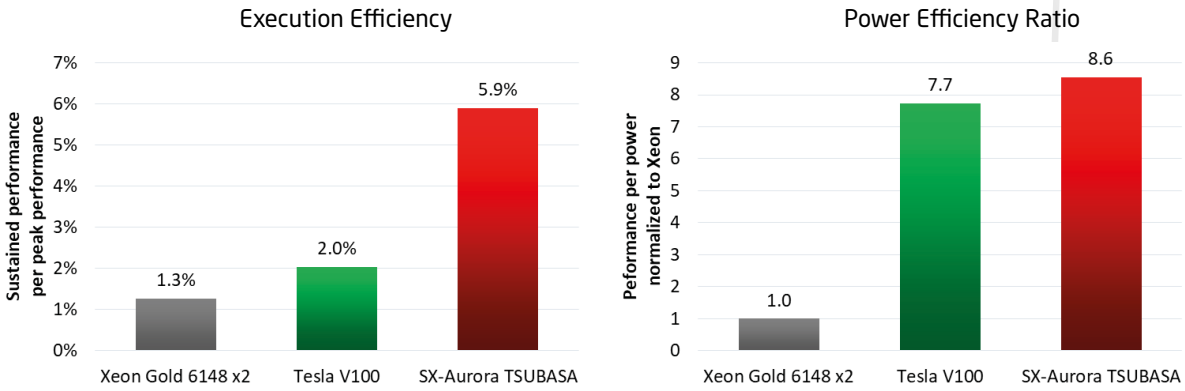
From very basic considerations it is clear that a vector architecture must be more power efficient than a scalar system. In a vector CPU, many actions are conducted for a large amount of data, not just for one operation on one number. Finally, this should have been one reason for the introduction of SIMD-instruction-sets by almost all scalar processor vendors.

In recent large installations, NEC SX-Aurora TSUBASA solutions could successfully provide its superiority in architecture and design just because of the power efficiency it provides. In some cases, NEC SX-Aurora TSUBASA even outperformed competing systems within a given power-envelope of an HPC-site. NEC SX-Aurora TSUBASA therefore not only excelled in terms of efficiency, but also on total computational capacity within a given power limit.

HPCG Benchmark

The High Performance Conjugate Gradient (HPCG) benchmark complements the TOP500-defining High Performance Linpack (HPL) as it uses real-world applications data access patterns, inverting a sparse matrix with a conjugate gradient algorithm that uses a symmetric Gauss-Seidel smoothed multigrid preconditioner. Opposed to HPL, which is a compute bound benchmark with performance close to the peak performance of a supercomputer, HPCG is a much more realistic benchmark with a memory bound problem and achieves only small fractions of the peak performance, reflecting the reality of most of today's HPC compute centers and their real world application performances.

As the following figure shows, due to the SX vector architecture and the 1.53TB/s high memory bandwidth per processor, SX-Aurora TSUBASA provides almost 6% of execution efficiency, and it is much higher efficiency compared to other architectures. SX-Aurora TSUBASA also provides the highest power efficiency between Xeon, Tesla, and SX-Aurora TSUBASA architectures.



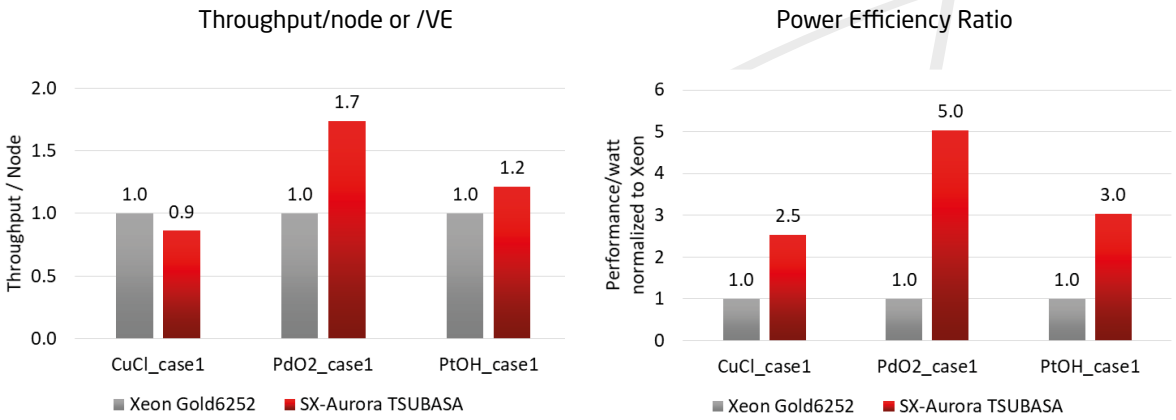
- **Xeon 6148:** Dual socket Xeon Gold 6148 (2 x 20 cores, 2.4GHz)
- **Tesla V100:** V100 x1 per host
- **SX-Aurora TSUBASA:** VE20B x1 per VH

Performance and power efficiency of HPCG on SX-Aurora TSUBASA

Ab Initio Theoretical Chemistry calculation - VASP

VASP (Vienna Ab initio Simulation Package, www.vasp.at), the well-known ab-initio theoretical chemistry application, is widely used by academia and industry. NEC has ported and optimized the VASP code for SX-Aurora TSUBASA. Adaptations due to the vector architecture have been necessary, straightforward. Coding for SX-Aurora TSUBASA is just based on standard languages and standard parallelization paradigms, which is a huge advantage compared to other computing and programming paradigms like GPU computing.

As a result, a single vector engine card VE type 20B of SX-Aurora TSUBASA provides 0.86 - 1.74 times the sustained performance compared to an x86 single node with two Intel Xeon Gold 6252 processors. When measuring the power consumption SX-Aurora TSUBASA provides even 2.5 - 5.0 times the power efficiency in terms of performance per Watt compared to the x86 system.



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

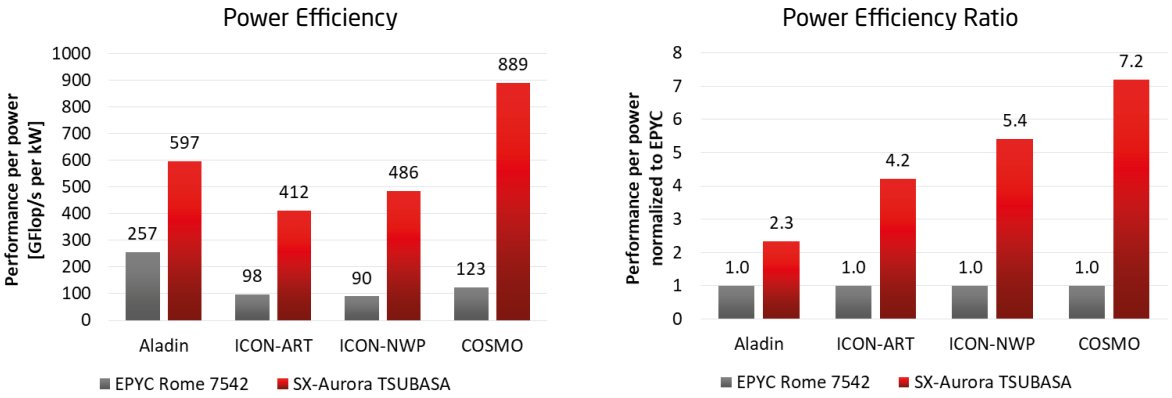
Intel Xeon Gold 6252
VASP: 5.4.4
Xeon: Gold 6252 (CLX Generation), 2 socket per node, 48 cores per node, 2.1GHz, air cooling

Performance and power efficiency of VASP on SX-Aurora TSUBASA

Meteorology

Meteorological sites very often require power efficiency because of their need for very big configurations while at the same time budgets for electricity and cooling are limited, as is the available power at their premises. Meteorological codes are mainly limited by memory bandwidth. Consequently, SX-Aurora TSUBASA has an advantage because it is the leading architecture in the market in this regard. With a very high computational efficiency, the system can reach a higher percentage of peak performance on the real application, which in turn results in improved power efficiency.

The following graph shows the "performance per Watt" comparison between SX-Aurora TSUBASA and a system based on AMD EPYC Rome. Each code, namely ALADIN, ICON-ART, ICON-NWP, and COSMO, is well known and frequently used in meteorology. On the HPL-benchmark, aka LINPACK, or TOP500, the AMD EPYC Rome system approximately provides the same power efficiency as SX-Aurora TSUBASA, but HPL is not a realistic code. On the real meteorological applications, the Aurora provides a 2.3 - 7.2 times higher power efficiency, which saves cost and enables bigger configurations.



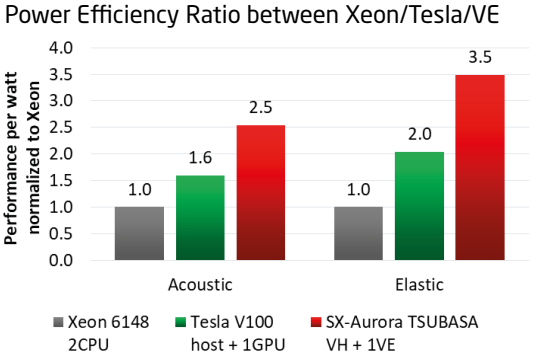
- **EPYC Rome 7542:** EPYC Rome 7542 32 cores/socket, 2.9GHz. 2 sockets per node
- **SX-Aurora TSUBASA:** VE10AE x8 / VH (single socket Rome)
- **ICON-ART:** Status as of 2019 for ICON-ART

Power efficiency of meteorology codes on SX-Aurora TSUABASA

Resource Exploration

The energy sector, especially resource exploration for oil and gas production, is another large field of application for HPC. An advantage in performance generates a real and immediate competitive advantage for the customer. At the same time, Oil & Gas companies also face limitations due to the ever-increasing power consumption of the HPC-equipment. The relevant codes crucially depend on memory bandwidth, and therefore NEC SX-Aurora TSUBASA provides an ideal platform for such applications. Again, the computational efficiency leads to a significant power-efficiency.

The following figure shows the ratio of power efficiency comparing Intel Xeon, NVIDIA Tesla, and SX-Aurora TSUBASA. The benchmark codes are an acoustic algorithm and an elastic algorithm, which are widely used in resource exploration. Again, a single VE of the SX-Aurora TSUBASA has the lowest peak performance among these three architectures, and it still provides the highest power efficiency.



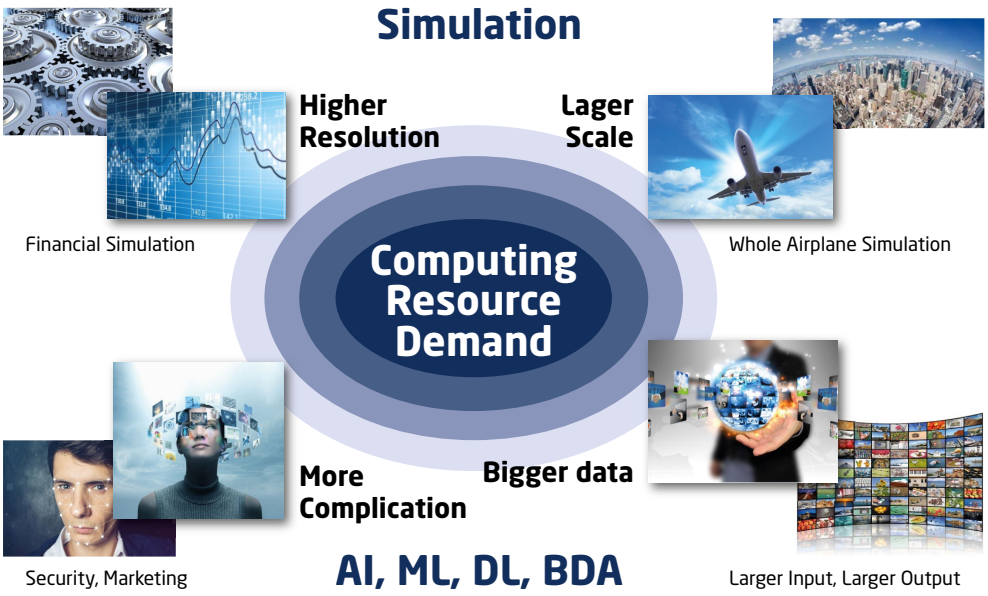
- **Xeon 6148:** Dual socket Xeon Gold 6148 (2 x 20 cores, 2.4GHz)
- **Tesla V100:** V100 x1 per host
- **SX-Aurora TSUBASA:** VE20B x1 per VH

Power efficiency of resource exploration algorithms on SX-Aurora TSUBASA

Conclusion of Application Performance

It is the real application performance that counts in HPC, not peak performance. It takes an efficient architecture both in terms of computational efficiency, i.e. application performance compared to peak performance, and power efficiency, because HPC-sites are hitting limits of power consumption and total cost of ownership. NEC SX-Aurora TSUBASA is a superior solution, and this has been proven on real applications.

NEC HPC Platform for Machine Learning and Deep Learning

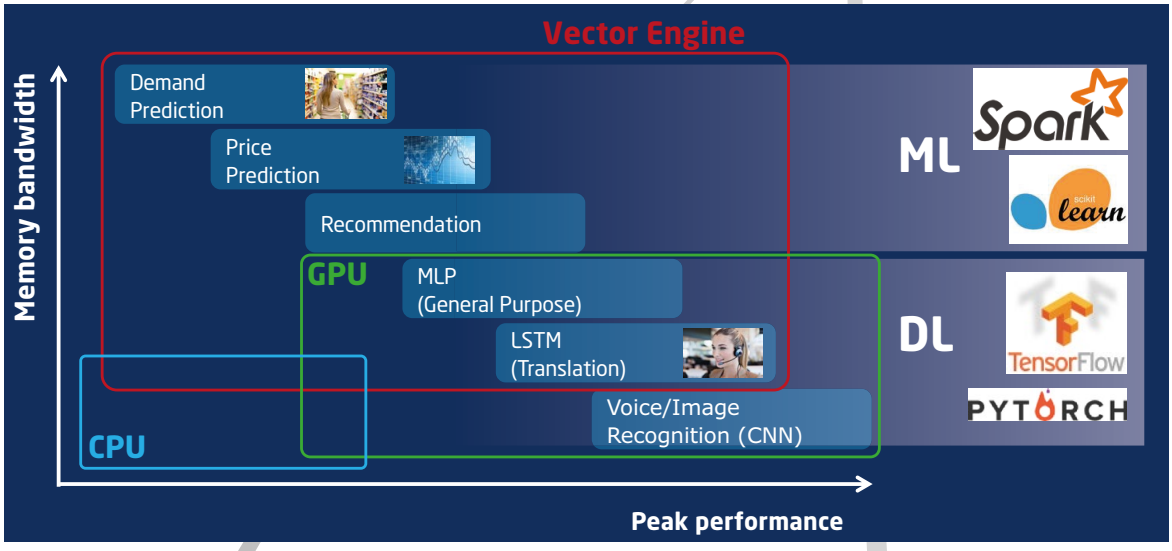


HPC capabilities have been used for scientific simulations to pursue higher simulation resolution, larger simulation scale, and therefore more accurate scientific result. Those HPC capabilities have contributed to scientific advancements through computation sciences. They have now started to be used for emerging technologies like Artificial Intelligence (A.I.), Machine Learning (ML), Deep Learning (DL), as well, with Big Data Analytics a vital ingredient of it all.

ML and DL have been used to empower business, research, and development competitiveness. These methods make suggestions and even decisions by processing huge amounts of data, therefore typical HPC capabilities are needed to get results faster, to handle larger data sets, and to produce results that are more accurate.

There are several applications in the ML and DL field, and those have different algorithmic characteristics. In the ML field, Apache Spark or Scikit-learn are widely used, often in applications that require a very high memory bandwidth. On the other hand, frameworks like TensorFlow or Pytorch are very famous in the DL field. Those applications require higher peak performance rather than memory bandwidth.

The Vector Engine processor of NEC SX-Aurora TSUBASA supercomputer excels in memory bandwidth because of an elaborate memory architecture based on six HBM2-modules per CPU. Due to these characteristics of the CPU, such a Vector Engine can accelerate applications of the ML field and a portion of the DL field.



ML/DL performance characteristics and each suited architectures

Machine Learning Acceleration by the NEC Frovedis Library

Frovedis (FRamework Of VEctorized and DIStributed data analytics) is an Open Source unified middleware implemented in C++ for NEC SX-Aurora TSUBASA. It covers a large part of Apache SPARK, implementing the same API for the domains Machine Learning, Matrix Libraries and Data Frame. Frovedis transparently supports multiple VE cards and VH servers by using MPI-based parallelism for scalable distributed processing.
Frovedis is very easy-to-use as the following figure shows. Only a changed import line in the source file and two added lines for starting and stopping the MPI based Frovedis service are required.

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

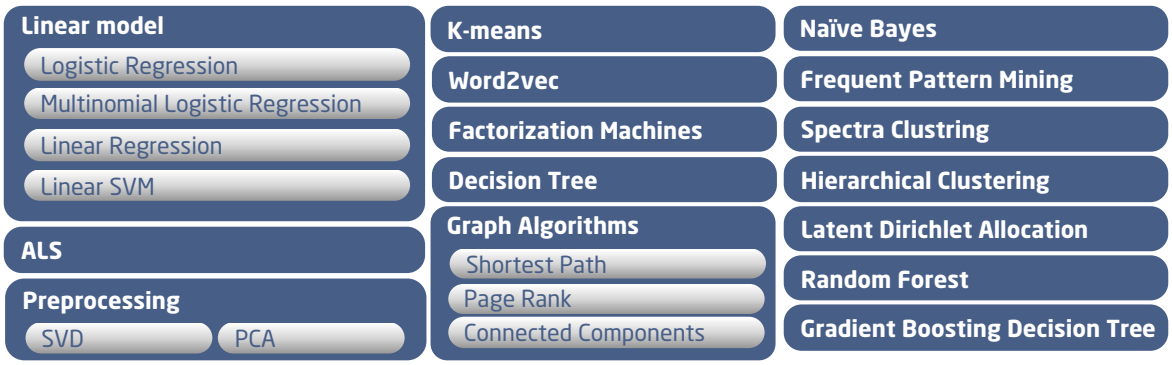
Start server

Stop server

Frovedis execution compared with the normal Spark execution

This framework has been developed to fully utilize the vector architecture and in particular take advantage of the superior bandwidth of the SX-Aurora TSUBASA. It supports the MLlib and scikit-learn libraries and achieves speedups of 10 to 100 in standard CPU versus VE comparisons.

The Frovedis Machine Learning Library is implemented with Frovedis Core and Matrix Libraries and supports both dense and sparse data. The huge bandwidth of VE contributes to the high performance of the sparse algorithms that are key to large scale ML. The figure below is a snapshot of the growing list of MLlib supported algorithms.

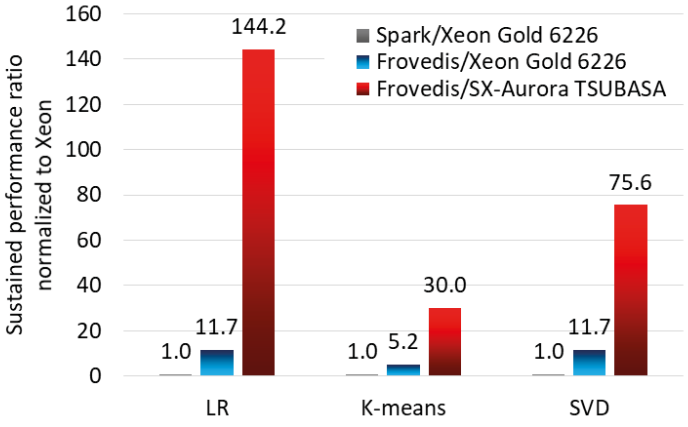


Snapshot of list of supported algorithms in the ML library

Examples of real world use cases of ML algorithms

Supported Algorithm	Type	Assumed Usecase
Linear Regression	Prediction	• Sales prediction • Supply and demand forecast
Naive Bayes (Multinomial)		
Logistic Regression (Binary, Mulit)	Classification	• Customer purchasing analysis • Recommend • Churn analysis • Credit screening • Fraud detection
SGD Classifier (Logistic Regression)		
Nearest Neighbors (k-Nearest Neighbors)		
Decision Tree (Classifier)		
PCA	Dimensionality reduction	• Understanding Customer needs • Fraud detection • Recommend
Latent Dirichlet Al Location (LDA)		
KMeans	Clustering	• Customer segment analysis

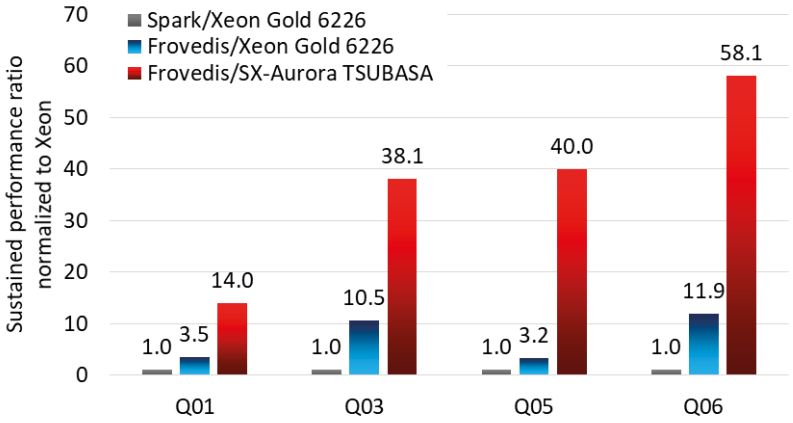
The following figure shows three examples from memory-intensive ML where the SX-Aurora TSUBASA vector architecture significantly increases the processing speed: (1) recommendation engines, (2) demand/price prediction, and (3) risk mitigation analysis.



- **Xeon Gold 6226:** 12 cores/socket, 2 socket/node, CentOS8.1
- **SX-Aurora TSUBASA:** A311-8, VE Type 10BE
- **Spark:** 3.0.0 for K-means and SVD, 2.2.1 for LR
- **Frovedis:** as of 2020/09/08

Sustained performance advantages of Frovedis on SX-Aurora TSUBASA

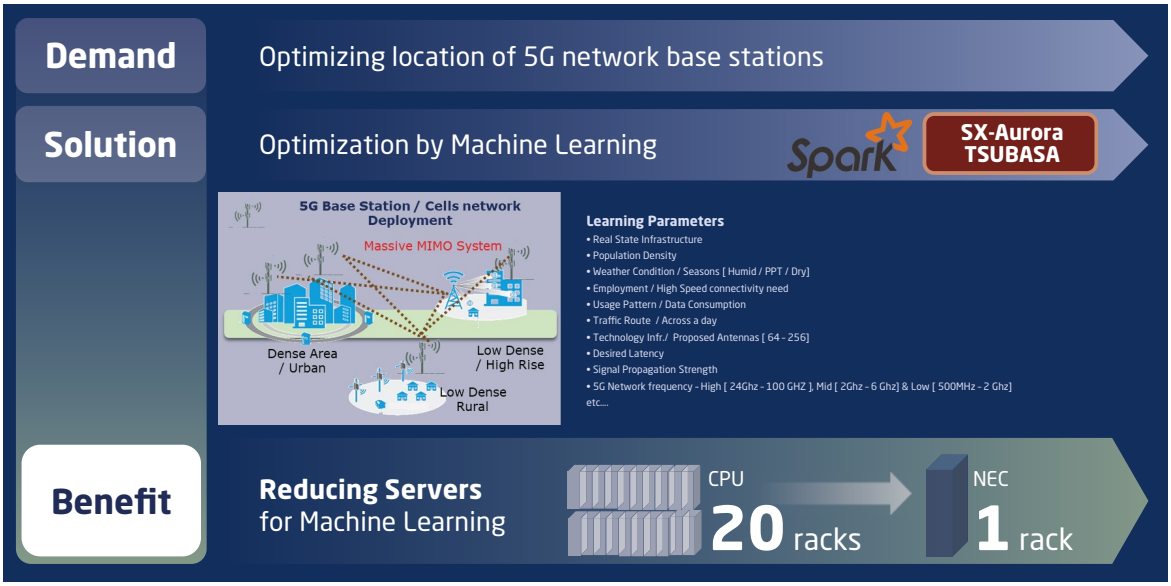
DataFrame performance is also accelerated significantly, as demonstrated in TPC-H SF-20 benchmark runs shown below. The test comparing again a Xeon Gold 6226 are Q01: group by/aggregate, Q03: filter, join, group by/aggregate, Q05: filter, join, group by/aggregate (larger join), and Q06: filter, group by/aggregate.



- **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

TPC-H SF-20 benchmark acceleration by Frovedis on SX-Aurora TSUBASA

One of the major fields for ML is "optimization". For example, ML-methods are applied to optimize each location of 5G network base stations used for telecommunication. Frovedis on SX-Aurora TSUBASA reaches a shorter time to solution, it can therefore reduce the amount of HPC equipment needed for operational ML, and consequently the user can save budget, space, electricity, thus operational and maintenance costs.



5G Network Optimization by Frovedis on SX-Aurora TSUBASA

NEC HPC platform for Deep Learning

The last few years have seen a growing interest in Deep Learning techniques, thanks to the major breakthroughs achieved in the fields of computer vision and natural language processing. These successes have encouraged researchers to explore many areas of application for such techniques. For example, even large computation tasks generally belonging to the high-performance computing domain, such as fluid mechanics, biology and astrophysics, may benefit from the application of Deep Learning to improve quality of simulations and increase computation performance.

At the core of Deep Learning, there is a set of algorithms called artificial neural networks (ANN), and their quick evolution has been propelled by the availability of a number of software frameworks that simplify development. NEC has been a pioneer in developing such frameworks, contributing to one of the first instances of such software: Torch. Currently, a number of alternatives have emerged, with PyTorch, TensorFlow, MXNet, CNTK being among the most popular deep learning frameworks.

Conceptually, these frameworks provide a ready-to-use set of neural network building blocks, and a high-level API to implement the neural network architecture. Writing a new neural network is as simple as sticking together a few lines of code using the python programming language. The frameworks will then take care of all the operations required to enable the training of the neural network and its execution. Given the high-level API, the user of the framework can ignore the low-level hardware details, focusing on the neural network algorithm design. However, the high-level API abstractions may also affect the overall efficiency of the implemented low-level computations. For instance, the high-level APIs generally expose neural network layers as atomic components of a neural network, requiring their execution to happen serially on the underlying hardware.

NEC Deep Learning Platform

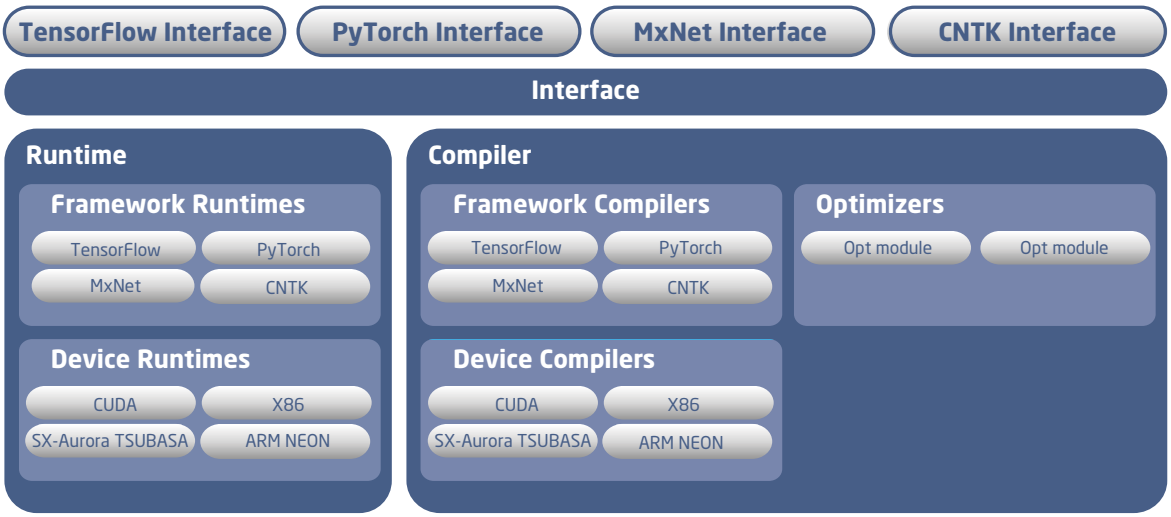
Building on its large experience in both AI applications and platforms, NEC designed a next generation deep learning platform aiming at portability, extendibility, usability and efficiency. The NEC's deep learning platform (DLP) seamlessly integrates into popular frameworks, rather than introducing "yet another API". As such, in contrast to other techniques, it does not replace any of the original functionality of the original deep learning frameworks but complements them.

Users continue writing neural networks using their favorite frameworks, e.g. TensorFlow or PyTorch. The NEC platform seamlessly analyses the neural network descriptions to provide additional hardware compatibility and improve performance. This step leverages the expertise of NEC in automated code generation techniques, allowing the system to create on the fly a computation library that is optimized specifically for the user's neural network. The generated code is compiled in implementations that are functionally equivalent for the user, performing the exact same set of computations on the data, but more efficiently. A code snippet that uses the NEC DLP looks as follows.

```
import framework

Import necdlp.framework as dlp

Model    = framework.ModelZoo.init("MyNeuralNetwork", …)

Model    = dlp.optimize(model)
Input    = (…load input…)

Result   = model(input)
```

NEC Deep Learning Platform: A code snippet

Two lines of code is everything that the user has to add to her deep learning code, in order to use NEC DLP and benefit from its optimizations. From an architectural perspective, NEC DLP is shown in the following figure.



NEC Deep Learning Platform Architecture

A peek into the optimization techniques

Deep Learning frameworks, like TensorFlow and PyTorch, execute their neural networks on a layer-by-layer basis, which can result in an inefficient use of the hardware executor's memory hierarchy and thereby in long execution times. NEC DLP addresses the problem by generating a new custom computation graph customized for the neural network at hand. The NEC platform: 1) analyzes the neural network structure to extract its computation graph; 2) then it performs transformation on this graph to generate more efficient computations taking into account the structure of the graph itself; 3) finally it generates computation libraries that are specific to the newly designed computation graph and to the target hardware. NEC DLP comes with a large set of optimization patters that can modify the network's structure, improve the reuse of data buffers, optimized the use of the device's memory hierarchy and computation units.

Neural Network Deployment

When a neural network model has been trained, it is supposed to be integrated into an application. Since models are developed using a Deep Learning framework, running the developed neural network requires shipping the entire framework together with the application. This is often too inefficient, in particular for some deployments where performance is critical. As such, often a developed neural network goes through expensive engineering efforts to port it into a stand-alone implementation that can be run independently from the original framework.

NEC DLP addresses this problem by supporting the deployment of the neural network models for any hardware supported by the framework. When required, NEC DLP can generate a software library that implements the neural network for the selected target device, e.g. NEC SX-Aurora TSUBASA. The deployment function relies on the same optimization engine and architecture of NEC DLP. The generated libraries only contain the neural network execution functions, parameters and a minimal set of helper functions, which significantly reduces the size of these libraries and simplifies their integration in the applications that need them.

Technical Specifications

SX-Aurora TSUBASA specifications

	Tower		Rackmount		Supercomputer	
Models						
Model name	A111-1	A101-1	A311-4	B300-8	B401-8	A511-64
Max. Vector Engines (VEs)	1	1	4	8	8	64
# of Vector Hosts (VHs)	1	1	1	1	1	8
Form Factor	Tower	Tower	1U rackmount	4U rackmount	2U rackmount	Proprietary rack
Vector Engine (VE)						
# of VEs	1	1	2, 4	8	8	64
Supported VE type	Type 10CE	Type 10B	Type 10BE	Type 20B	Type 20A Type 20B	Type 10AE Type 10BE
Max. VE performance (TF)	2.15	2.15	8.65	19.66	24.57	155.71
Max. VE memory bandwidth (TB/s)	1.00	1.22	5.40	12.28	12.28	86.50
Max. VE memory capacity (GB)	24	48	192	384	384	3,072
Vector Host (VH)						
Processors/VH	1	1	2	2	1	2
Processor	Intel® Xeon® Scalable Processor				AMD EPYC™ Processor	Intel® Xeon® Scalable Processor
Max. memory configuration	DDR4 DIMM x 6 / CPU				DDR4 DIMM x 8 / CPU	DDR4 DIMM x 6 / CPU
Max. memory capacity (GB)/VH	96	96	192	192	512	192
OS	Red Hat Enterprise Linux 8.1 - 8.x / CentOS 8.1 - 8.x					
Interconnect						
InfiniBand	-	-	EDR/HDR100	HDR100	HDR200	EDR/HDR100
Max. HCAs (InfiniBand EDR)	-	-	2	2	2	16
Bandwidth per direction (GB/s)	-	-	25	25	50	200
Power and Cooling						
Power consumption (HPL)	0.6 kW	0.6 kW	1.9 kW	3.3 kW	< 3.3 kW	< 28.0 kW
Cooling	Air	Air	Air	Air	Water + Air	Water + Air
Software						
Bundled software	VE controlling software, VE driver					
Software developers kit	Vector compiler/debugger/libraries/profiler for VE					
MPI	MPI library for VE					

Vector Engine (VE) Specifications

	Type 10B	Type 10CE	Type 10BE	Type 10AE	Type 20B	Type 20A
Core Specifications						
Clock speed (GHz)	1.400	1.400	1.408	1.584	1.600	1.600
Performance (GF)	268.8	268.8	270.3	304.1	307.2	307.2
Average memory bandwidth (GB/s)	153.6	125.0	168.9	168.9	192.0	153.6
Processor Specifications						
# of cores / processor	8	8	8	8	8	10
Peak performance (TF)	2.15	2.15	2.16	2.43	2.45	3.07
Memory bandwidth (TB/s)	1.22	1.00	1.35	1.35	1.53	1.53
Cache capacity (MB)	16	16	16	16	16	16
Memeory capacity (GB)	48	24	48	48	48	48

NEC Corporation
Headquarters, NEC Supertower,
5-7-1, Shiba, Minato-ku, Tokyo
Japan
email: info@hpc.jp.nec.com

www.nec.com

NEC Deutschland GmbH
HPC EMEA Headquarter
Fritz-Vomfelde-Straße 14-16
D-40547 Düsseldorf
Tel.: +49 (0) 211 5369 0
email: info@nec.de

HPC Technology Center
Raiffeisenstraße 14
D-70771 Leinfelden-Echterdingen
Tel.: +49 (0) 711 78 055 0

HPCE France Division
3 Parc Ariane
Immeuble Saturne
F-78284 Guyancourt
Tel.: +33 (0) 139 30 66 000

HPCE UK Division
NEC Europe Ltd
Athene, Odyssey Business Park
West End Road, South Ruislip, Middlesex, HA4 6QE, GB
DDI: +44 (0) 7876 651495

NEC Laboratories Europe GmbH
Kurfürsten-Anlage 36
D-69115 Heidelberg
Tel.: +49 (0) 621 4342-0
email: hdoffice@neclab.eu

Aurora Forum
www.hpc.nec