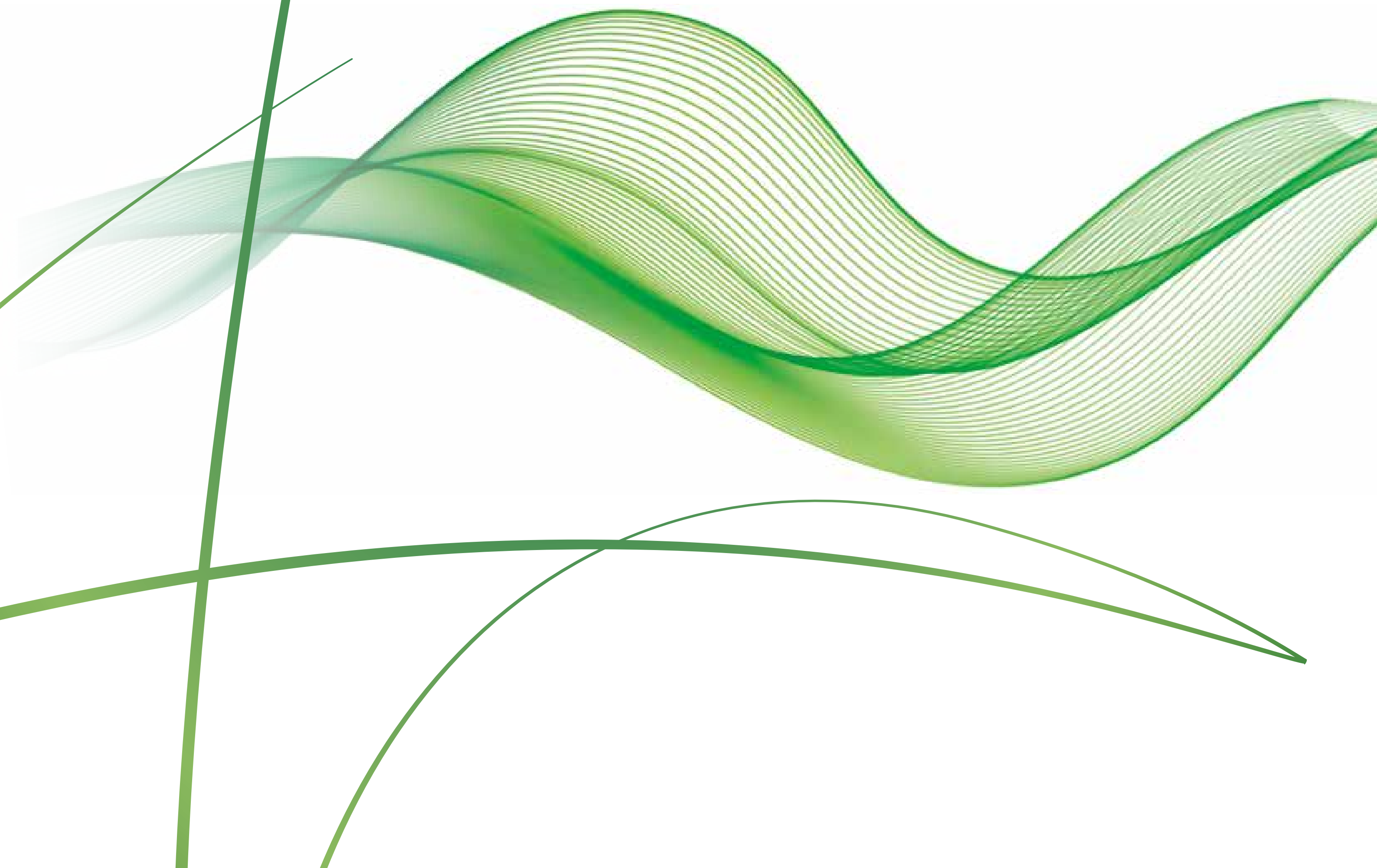
\Orchestrating a brighter world

NEC

HPC Companion 2019/20

Technology Guide



HPC Companion 2019/20

Technology Guide

Table of Contents

NEC as a driver of innovation

Supercomputing and Aritificial Inteligence Made in Japan

High Performance Computing today is many-faceted: it is computational power at its extreme, populating the largest datacenters around the world, utilized for leading-edge research and development by the world's most renowned academic and research facilities. But it has also become mainstream: it is now firmly established itself as the third pillar of science and research alongside theory and experiment.

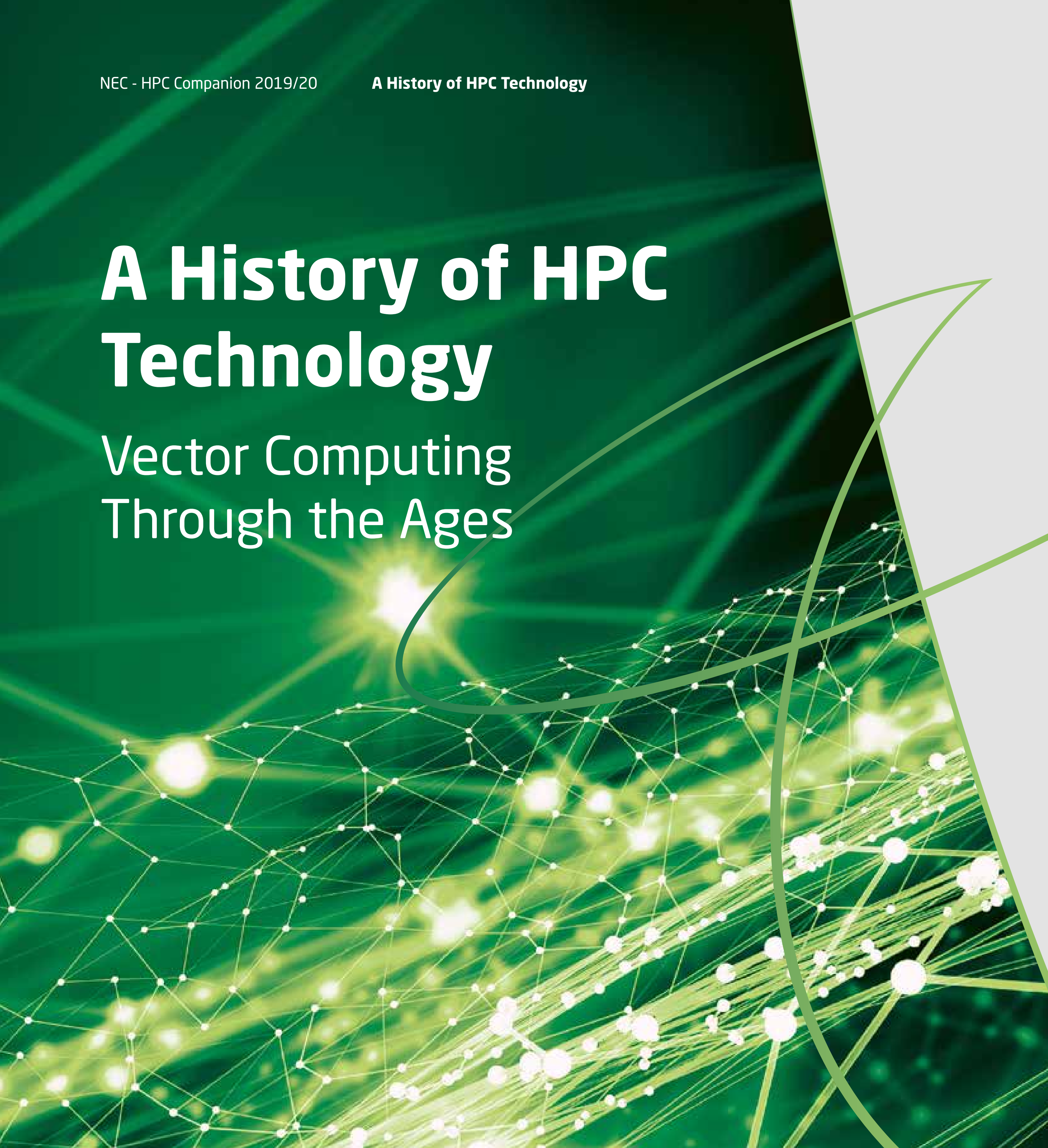
Artificial intelligence has now experienced a reincarnation, going by the name of "deep learning". Old and well-designed algorithms not only have been improved, but can now deliver to their full strength because the computational capacity provided by the current HPC systems turns their theoretical potential into full predictive power! NEC is a master of HPC, and has been so for decades. We know exactly what our customers from automotive or engineering, medical and life sciences or astrophysical, meteorological and climate research need for optimizing their productive workflow.

But NEC is also a master of Artificial Intelligence. In our Research Labs in Europe, the U.S. and around the world, algorithms are being designed and software is being developed to exploit the full computational power of NEC's vector engine SX-Aurora TSUBASA, designed with highest memory bandwidth and highest sustained application performance in mind.

NEC's solution portfolio for HPC and A.I. comprised not only computational hardware capacity, but also HPC storage solutions based on parallel filesystems commonly used throughout the HPC world. With NEC's deployment solution LXC3-neo, managing a full-fledged HPC environment is made as easy as possible. Our expertise in code and performance optimization, and the excellence of our presales and benchmarking teams is renowned. Our system integration and support team is highly experienced in even the most challenging configurations and a paradigm of customer orientation.

NEC combines Japanese leading-edge technology with a European spirit. As a comprehensive provider of HPC solutions, we strive after pushing the boundaries of all human achievability, enabling people to live brighter lives. We summarize this approach in our business message: Orchestrating a brighter world.

Yuichi Kojima
Managing Director
NEC Deutschland GmbH
Vice President High Performance
Computing Europe



A History of HPC Technology

Vector Computing Through the Ages

During the last years x86 systems dominated the HPC market, with GPUs and for some time many-core-systems making their inroads. But this period is coming to an end, and it is again time for a differentiated complementary HPC targeted system.

NEC has now developed such a system based on over 30 years of experience, retaining the virtues of traditional vector systems, in particular high memory bandwidth, combining this with a brand new innovative concept.

This article reviews the past developments of HPC technology, analyzes the status quo, and briefly describes NEC's view on the future and related products.

History of HPC, and What We Learn

The Early Days

When did HPC really start? Probably one would think about the time when Seymour Cray developed high-end machines at CDC and then started his own company Cray Research, with the first product being the Cray-1 in 1976. At that time supercomputing was very special, systems were expensive, and access was extremely limited for researchers and students. But the business case was obvious in many fields, scientifically as well as commercially.

These Cray machines were vector computers, initially featuring only one CPU, as parallelization was far from being mainstream, even hardly a matter of research. And code-development was easier than today, a code of 20,000 lines was already considered “complex” at that time, and there were not so many standard codes and algorithms available anyway. For this reason code development could adapt to any kind of architecture to achieve optimal performance.

In 1981 NEC started marketing its first vector supercomputer, the SX-2, a single-CPU-vector-architecture like the Cray-1. The SX-2 was a milestone, it was the first CPU to exceed one GFlops of peak performance. Also Fujitsu and Hitachi were actively pursuing this direction.

The Attack of the "Killer-Micros"

In the 1990s the situation changed drastically, and what we see today is a direct consequence. The "killer-micros" took over, new parallel, later "massively parallel" systems (MPP) were built based on microprocessor architectures which were predominantly targeted to commercial sectors, or in the case of Intel, AMD or Motorola to PCs, to products which were successfully invading the daily life, partially more than even experts had anticipated. Clearly one could save a lot of cost by using, in the worst case slightly adapting these CPUs for high-end-systems, and by focusing investments on other ingredients, like interconnect technology. At the same time increasingly LINUX was adopted and offered a cheap generally available software platform to allow everybody to use HPC for a reasonable price.

These systems were cheaper because of two reasons: the development of standard CPUs was sponsored by different businesses, and another expensive part, the memory architecture with many CPUs addressing the same flat memory space, was replaced by systems comprised of many distinct so-called nodes, complete systems by themselves, plus some kind of interconnect technology, which could also be taken from standard equipment partially.

The impact on the user was the necessity to code for these "distributed memory systems". Standards for coding evolved, PVM and later MPI, plus some proprietary paradigms. MPI is dominating still today.

In the high-end the special systems were still dominating, because they had architectural advantages and the codes were running with very good efficiency on such systems. Most codes were not parallel at all or at least not well parallelized, let alone on systems with distributed memory. The skills and experiences to generate parallel versions of codes were lacking in those companies selling high-end-machines and also on the customers' side. So it took some time for the MPPs to take over. But it was inevitable, because this situation, sometimes called "democratization of HPC", was positive for both the commercial and the scientific developments. From that time on researchers and students in academia and industry got sufficient access to resources they could not have dreamt of before.

The x86 Dominance

One could dispute whether the x86 architecture is the most advanced or elegant, undoubtedly it became the dominant one. It was developed for mass markets, this was the strategy, and HPC was a side effect, sponsored by the mass market. This worked successfully and the proportion of x86 LINUX clusters in the Top500-list increased year by year.

Other architectures experienced economical pressure, not only the vector systems, also competing scalar concepts. The HP-precision-architecture is basically gone, MIPS is mostly used for controllers, SPARC is still alive at Oracle and Fujitsu, but it is doubtful that it will recover the development cost. The Motorola-architecture is gone, other microprocessors vanished as well. IBM Power is still alive, but it is at least hard to predict its future. On the vector side Cray stopped the products lines, Fujitsu and Hitachi bailed out, NEC is the only remaining vendor, but committed to the future.

Because of the mass markets it economically made sense to invest a lot into the development of LSI technology. The related progress, which led to more and more transistors on the chips and increasing frequencies, helped HPC applications, which got more, faster and cheaper systems with each generation. Software did not have to adapt a lot between generations, and the software developers rather initially focused on the parallelization for distributed memory, later on new functionalities and features. This worked well for quite some years.

Systems of sufficient size to produce relevant results for typical applications are cheap, and most codes do not really scale to a level of parallelism to use thousands of processes. Some lighthouse projects do, after a lot of development which is not easily affordable, but the average researcher or student is often completely satisfied with something between a few and perhaps a few hundred nodes of a standard LINUX-cluster. Just to put it into perspective, the LINUX-virtual-machine on my Windows laptop is faster than the Cray-X-MP which I used for my thesis.

In the meantime already a typical Intel based dual socket node of a standard cluster has 32 or more cores, this number is still growing. Therefore parallelization is increasingly important. There is “Moore’s law”, invented by Gordon E. Moore, a co-founder of the Intel Corporation, in 1965. Originally it stated that the number of transistors in an integrated circuit doubles about every 18 months, leading to an exponential increase. This statement was often misinterpreted that the performance of a CPU would double every 18 months, which is misleading. We observe ever increasing core counts, the transistors are just there, so why not? That applications might not always be able to use these many cores is another question, in principle the potential performance is there.

The "standard benchmark" in the HPC-scene is the so-called HPL, High Performance LINPACK, and the famous Top500 list is based on this. This HPL benchmark is only one, and in fact the latest, of three versions of the original LINPACK-benchmark, this is often forgotten! The other pieces, matrix dimension 100 and 1,000, are rather testing compiler technology and the performance of individual cores, memory latency etc., not scalability, and therefore did not show such a favourable progress in performance. So these small cases could not serve to spawn competition for prestige even between nations.

There are at most a few real codes in the world with similar characteristics as the HPL. In other words: it does not tell a lot. This is a highly political situation and probably quite some money is wasted which would better be invested in brainware.

It is increasingly doubtful that LSI technology will advance at the same pace in the future. The lattice constant of Silicon is 5.43Å. Nowadays typical is a 7nm process. This scale is about 25 times the lattice constant. Some scientists assume there is still room to improve even more, but once the features on the LSI have a thickness of only a few atom layers one cannot get another factor of ten. Sometimes the impact of some future "disruptive technology" is claimed ... there will be ways, there will be limits.

Another obstacle is the cost to build a chip-fab for a new generation of LSI technology. Certainly there are ways to optimize both production and cost, and there will be bright ideas, but with current generations a fab costs at least many billion $. A slow down of the product innovation cycle, no more tick-tock, is necessary to recover the investment.

Memory Bandwidth

For HPC-applications it is mostly not the compute power of the CPU which counts, it is the speed with which the CPU can exchange data with the memory, the so-called memory bandwidth. If it is neglected the CPUs will wait for data, the performance will decrease. To describe the balance between CPU performance and memory bandwidth one often uses the ratio between operations and the amount of bytes that can be transferred while they are running, the "byte-per-flop-ratio". Traditionally vector machines were always superior in this regard, and this is also one design goal of NEC's new machine. For this reason a vector machine can realize a higher fraction of its theoretical performance on a real application than a scalar machine, the "efficiency".

Power Consumption

One other but really overwhelming problem is the increasing power consumption of systems. A dual-socket-node, depending on how it is equipped with additional features, will consume between 400 Watt and 50 Watt under normal production workload. This only slightly varies between different generations, but seems to increase slowly. Even in Europe sites are talking about electricity budgets in excess of 1 MWatt, even beyond 10 MWatts. One also has to cool the system, which adds something like 10% - 30% to the electricity cost! The cost to run such a system over 5 years will be in the same order of magnitude as the purchase price!

The power consumption will grow with the frequency of the CPU, this is the dominant reason why frequencies are at least stagnating, rather decreasing between product generations. There are more cores, but individual cores of an architecture are getting slower, not faster! This has to be compensated on the application side by increased scalability, which is not always easy to achieve.
In the past users easily got a higher performance for almost every application with each new generation, this is no longer the case.

Architecture Development

So the basic facts clearly indicate limitations of the LSI technology, and they are known since quite some time. Consequently this leads to attempts to get additional performance by other means. And the developments in the market clearly indicate that high-end users are willing to consider the adaptation of codes.

To utilize available transistors CPUs with more complex functionality are designed, utilizing SIMD-instructions or providing more operations per cycle than just an add and a multiply. SIMD is “Single Instruction Multiple Data”, and it is an idea with quite a history, “Thinking Machines” was founded in the 80s. At Intel the SIMD-instructions initially provided two results in 64-bit-precision per cycle rather than one, in the meantime with AVX it is even four, on recently with AVX512 already eight.

This is still a moderate level of parallelism, but it makes a lot of sense. The electricity consumed by the “administration” of the operations, code fetch and decode, configuration of datapaths on the CPU etc. is a significant portion of the power consumption. If the energy is used not only for one mathematical operation but for two, four, or even eight, the energy per operation is obviously reduced.

In essence, and in order to overcome the performance bottleneck, the architectures developed into the direction which to a larger extent was already realized in the vector supercomputers decades ago! There is almost no architecture left without such features. But SIMD-operations require application codes to be vectorized. During the days of purely scalar computing code developments and programming paradigms did not care for fine-grain data parallelism on scalar CPUs, which is the basis of applicability of SIMD. But now, strictly speaking, there is no scalar CPU left, standard x86-systems do not achieve their optimal performance on purely scalar code as well!

Alternative Architectures?

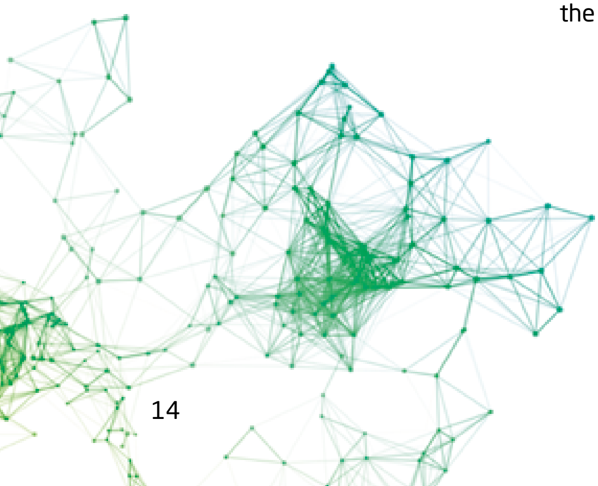
During recent years GPUs and many-core-architectures are increasingly used in the HPC market. Although these systems are somewhat diverse in detail, in any case the degree of parallelism needed is much higher than in standard CPUs. This points into the same direction, a low-level parallelism is used to address the bottlenecks. In the case of NVIDIA GPUs the individual so-called CUDA-cores, relatively simple entities, are steered by a more complex central processor, realizing a high level of SIMD-parallelism.

So these alternative architectures not only develop into the same direction, the utilization of fine-grain parallelism, they also indicate that coding styles which suite real vector machines will dominate the future software developments. Note that at last some of these systems have an economic backing by mass markets, gaming! But other mass markets are looming, in particular deep learning and data analytics.

Lessons learned

The standard LINUX-clusters or other systems, which are based on standard technologies, are providing a great tool for science and engineering, and are available to many users. In a lot of cases these systems are sufficient for the job. Consequently NEC has to provide such systems also in the future, and in fact this is a growing business since the late 90s.

In addition, NEC has developed an alternative architecture, the SX-Aurora TSUBASA, to tackle the bottlenecks which were described in previous chapters. By the way, TSUBASA is Japanese for “wing”.



The business model must be adapted. A system cannot be successful if it is only available in the form of high-end installations. The GPUs and many-core-systems have shown that researchers and students are willing to adapt their projects to a differentiated architecture if it can achieve results, performance or performance per price, which are not possible otherwise.

To make it easy to use and to make the user feel comfortable from the first login the vector-system needs to be closely integrated with what the normal user knows well, specifically a LINUX-based environment.

At the same time NEC must make sure that it will be easy to get a good efficiency from the code porting, it should not be as complicated as CUDA programming, and in fact because of the experience with vector compilers NEC has a great advantage here.

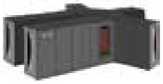
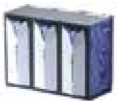
SX-Aurora TSUBASA

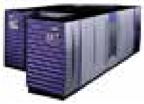
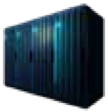
The New Generation NEC Vector Supercomputer

In early 2018 NEC launched the new generation of the SX vector architecture, SX-Aurora TSUBASA, offering a hitherto unparalleled performance especially for memory-bound applications. This is what the new Aurora vector engine is about.

SX-Aurora TSUBASA

The SX Vector Architecture from Past to Future

Type	Year	Tech-nology	CPU Frequency	CPU Perfor-mance	CPU Memory Bandwidth
SX-2	1983	Bipolar	166 MHz	1.3 GFlops	10.7 GB/sec
SX-3	1989	Bipolar	340 MHz	5.5 GFlops	12.8 GB/sec
SX-4 	1994	350 nm	125 Mhz	2.0 GFlops	16.0 GB/sec
SX-5	1998	250 nm	250 MHz	8.0 GFlops	64.0 GB/sec
SX-6 	2001	150 nm	500 MHz	8.0 GFlops	32.0 GB/sec

Type	Year	Tech-nology	CPU Frequency	CPU Perfor-mance	CPU Memory Bandwidth
SX-7 	2002	150 nm	552 MHz	8.8 GFlops	35.3 GB/sec
SX-8	2004	90 nm	1.0 GHz	16.0 GFlops	64.0 GB/sec
SX-9	2007	65 nm	3.2 GHz	102.4 GFlops	256.0 GB/sec
SX-ACE	2013	28 nm	1.0 GHz	256.0 GFlops	256.0 GB/sec
SX-Aurora TSUBASA 	2018	16 nm	1.4-1.6 GHz	2,150 - 2,457 GFlops	1,228.8 GB/sec

As already stated NEC started marketing the first commercially available generation of this series, the SX-2, in 1983. First systems sold in Europe were SX-3, shared memory machines with up to four very strong processors. The success was continued with the SX-4, with up to 32 strong CPUs on a shared memory, successfully competing against the Cray T90.

With the SX-5 this direction was continued, it was a very successful system pushing the limits of CMOS-based-CPUs and multi-node-configurations. At that time slowly the efforts on the users' side to parallelize their codes using "message passing", mostly PVM and MPI at that time, started to surface, so the need for a huge shared memory system slowly disappeared. Still there were quite some important codes in the market which were only parallel using a shared-memory-paradigm, initially vendor-specific "microtasking" or "autotasking". NEC and some other vendors could automatically parallelize code, and later the standard OpenMP. So there was still sufficient need for a huge and highly capable shared-memory-system.

Because of continued software development more and more codes could utilize distributed memory systems, and moreover the memory architecture of huge shared memory systems became increasingly complex and that way costly.
A memory is made of "banks", individual pieces of memory which are used in parallel. In order to provide the data at the necessary speed for a CPU a certain minimum amount of banks per CPU is required, because a bank needs some cycles to recover after every access. Consequently the number of banks had to scale with the number of CPUs, so that the complexity of the network connecting CPUs and banks was growing with the square of the number of CPUs. And if the CPUs became stronger they needed more banks, which made it even worse. So this concept became too expensive.

Already at that time NEC had a lot of experience in mixing MPI- and OpenMP-parallelization in applications, and it turned out that for the very fast CPUs the shared-memory parallelization was typically not really optimal for more than four or eight threads. This number is no basic law of nature, and surely it depends on the individual capability of the CPUs.

NEC adapted the concept accordingly, which lead to the SX-6 and the famous Earth-Simulator, which kept the #1-spot of the Top500 for some years. The Earth-Simulator was a different machine, but based on the same technology level and architecture as the SX-6. These systems were the first implementation ever of a vector CPU on one single chip. The CPU by itself was not the challenge,

but the interface to the memory to support the bandwidth needed for this chip, basically the "pins". This kind of packaging technology was always a strong point in NEC's technology portfolio. NEC developed an own interconnect technology, the so-called IXS, a fully non-blocking crossbar. Based on this clusters were built with a log of individual SX-nodes, which by themselves were powerful systems already.

The SX-7 was a somewhat special machine, it was based on the SX-6-technology-level, but featured 32 CPUs on one shared memory.

The SX-8 then was a new implementation again with some advanced features which directly addressed the necessities of certain code structures frequently encountered in applications. The communication between the worldwide organization of application analysts and the hard- and software developers in Japan had proven very fruitful, the SX-8 was very successful in the market.

With the SX-9 the number of CPUs per node was pushed up again to 16 because of certain demands in the market. Just as an indication for the complexity of the memory architecture: the system had 16,384 banks! The peak performance of one node was 1.6 TFlops. Keep in mind that the efficiency achieved on real applications was still in the range of 10% in bad cases, up to 20% or even more, depending on the code. One should not compare this peak performance with some standard chip today, this would be totally misleading.

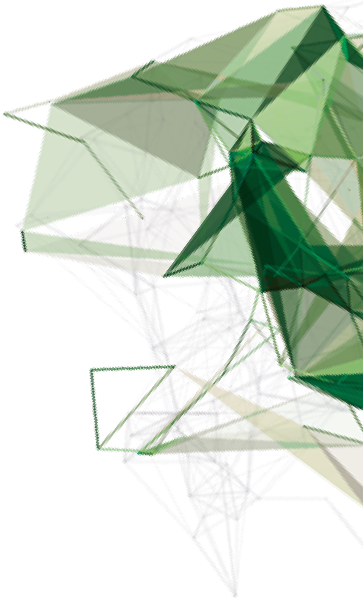
In November 2014 NEC announced the next generation of the SX-vector product, the SX-ACE. The main design target was the reduction of power consumption per sustained performance, and measurements have proven this has been achieved. In order to do so and in line with the observation that a shared-memory-parallelization with vector CPUs will be useful on four cores, perhaps eight, but normally not beyond that, the individual node consists of a single-chip-four-core-processor, which inherits memory controllers and the interface to the communication interconnect, a next generation of the proprietary IXS. That way a node is a small board, leading to a very compact design. The complicated network between the memory and the CPUs, which took about 70% of the power consumption of an SX-9, was eliminated. And naturally a new generation of LSI and PCB technology leads to a reduction of both power consumption and cost.

The first incarnation of a completely refurbished product line is available now, the NEC SX-Aurora TSUBASA. At times of stagnating core performance of standard components, the NEC system is a clear alternative for quite some application fields.

Aspects to address, lessons learned

NEC has learned the lessons from the past, and the new product addresses these quite distinctively:

- The architecture of the SX-line was developed in the late 1970s, early 1980s. Imagine that at the times of SX-2 a simple add pipe filled a complete rack. The LSI technology today obviously allows for a much higher level of integration, which makes it possible to adapt the architecture to become much more complex. NEC make use of these advantages. The machine features a vector architecture, conceptually very similar to the former SX-line, but with a lot of enhancements.
- The SX was using big endian. When it was first implemented this was o.k., there were even proprietary formats (Cray, IBM, VAX), and for IEEE it was by no means clear whether little or big endian was going to dominate. Nowadays it is very clear because of the dominance of x86, and one has to assume big endian will disappear. So the new vector system of NEC features little endian, which makes a tight integration into surrounding environments consisting of x86 or ARM much simpler.
- The SX-line was controlled by SUPER-UX, a proprietary UNIX. As much as this was appropriate in the past to provide a stable production quality environment, today Linux is the de facto standard. Users do not want to cope with other UNIX-breeds any more, and they are used to the tools that are available even on the laptop. So for the user to accept a new system we provide a well-familiar Linux environment, and the quasi-standard tools.
- Since many years NEC is enhancing the level of integration of SX-machines into the surrounding environment. The first steps taken already years ago were the provision of a fast parallel shared filesystem to both vector and scalar nodes and the availability of a highly advanced workload manager which could handle even workflows, sequences of jobs with logical dependency, on these different components. The natural extension is the provision of an MPI which can utilize both vector and scalar nodes in one application, for example for coupled models in climatology. NEC is pushing this kind of integration to much higher levels now for a coarse granular hybrid system.

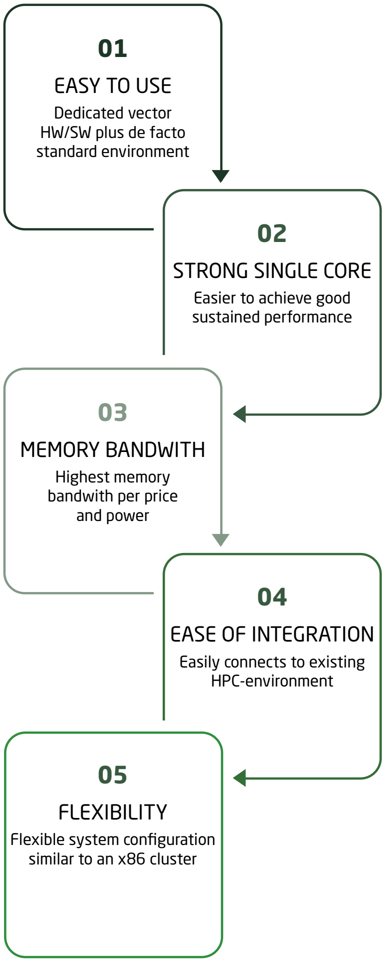


- Traditionally NEC has always been very successful in quite some vertical markets, namely high-end university sites, in automotive and aerospace, and in meteorology and climatology. And in addition, NEC now explores new vertical markets, as vector processing continues to be the optimal target architecture for many emerging application fields as well. For example in “deep learning” quite some algorithms rely on “sparse matrix times vector”, a bandwidth-limited kind of operation for which already the current SX-ACE can achieve remarkable performance and, much more important, industry leading power efficiency in terms of GFlops per Watt, as it shows in the HPCG-benchmark. It is fair to assume that quite some basic algorithmic constructs which were well known in HPC since many years will show up in many other fields that were traditionally not addressed by SX.
- NEC’s SX-system targeted high-end computing in the past, fully blown configurations were very expensive, and even small configurations exceeded the available budgets of individual researchers or small institutes. NEC’s new product is designed to provide a solution already at a very low price level. Since a lot of code development is originating from smaller institutions or even from individual scientists or engineers, often as a community effort, the SX-systems were not easily available to code developers and therefore not considered a platform to support. So NEC will intensify the contact to code development people of the present and the future, students, researchers and professional coders, providing them with a full-fledged software development environment comprising high-quality compilers and libraries.

NEC addresses all these aspects with the SX-Aurora TSUBASA.

Basic Product Strategy

It is easy to deduct the necessary features of the refurbished product line from the lessons learned in the past combined with the opportunities resulting from the advanced chip design technologies, summarized in one viewgraph:


01
EASY TO USE
Dedicated vector HW/SW plus de facto standard environment
02
STRONG SINGLE CORE
Easier to achieve good sustained performance
03
MEMORY BANDWITH
Highest memory bandwith per price and power
04
EASE OF INTEGRATION
Easily connects to existing HPC-environment
05
FLEXIBILITY
Flexible system configuration similar to an x86 cluster

Ease of Use

The user works in a complete Linux environment. There is nothing specific except compilers and highly tuned libraries. An execution can be started on the Linux system either interactively or via a batch system. In principle everything that would compile for a Linux system also compiles for the vector system. The compilers support Fortran 2003, some extensions of Fortran 2008, and C++14. The compilers are able to vectorize and auto-parallelize loops. We support OpenMP and MPI for manual parallelization. With 30 years of experience with vector compilers NEC certainly knows how to do that.

Superior Single Core Performance

As already emphasized, memory bandwidth is the key ingredient to providing a strong core for real application performance. Low computational efficiency because of lacking access to memory is a waste of energy.

But the current trend in the HPC arena, increasing numbers of cores keeping or even reducing the clock frequency, will not solve this problem. After years of surviving on the progress of higher chip integration, now is the time for an evolution of hardware architectures to achieve higher computational efficiency beyond just a few percent. NEC contributes to this and leads the market of vector supercomputing, achieving high efficiency computing.

The vector-architecture does not only address memory bandwidth, it is also a means coping with ever increasing latencies, in particular the latency of memory access. Latency hiding will be a dominating topic in the future, and a vector architecture is a natural way to address it.

Superior Memory Bandwidth

Memory bandwidth is and always has been a hot topic in the HPC market. At all times everybody was aware of the fact that the balance between computational power of a core and available memory bandwidth for this core is of high importance, and in the course of time this balance became worse. NEC leads the market in terms of memory bandwidth per CPU and even more per core, because this is the indispensable prerequisite for good efficiency for real world computational workload.

The new system features the best memory bandwidth, but even more so the best memory bandwidth per price and per power consumption. Consequently the system achieves very good efficiency with low power consumption.

System Integration

Installations will be modular in the future, they will consist of connected clusters with different node architectures. There will be x86-based systems, dual- or more sockets, systems with additional accelerators and GPGPUs, strongly connected systems for “scale-up” and weakly connected for throughput workload.

Flexibility

The latest product line of vector systems allows for very flexible configurations, from a deskside system up to a very dense high-end configuration. In particular, it can be configured as a coarse granular hybrid system, a mixture of purely scalar nodes and nodes, which contain vector CPUs, which we call “vector engines” (VE). That way the system can be tailored to every need depending on the nature of the applications and the workload to be processed.

This all implies that a new system needs to fully adapt to all kinds of surroundings, be compatible as much as possible and provide a user environment that people find even on their laptop or deskside x86-workstation.

NEC’s new product is designed to fulfill this requirement.

Hybrid Computing

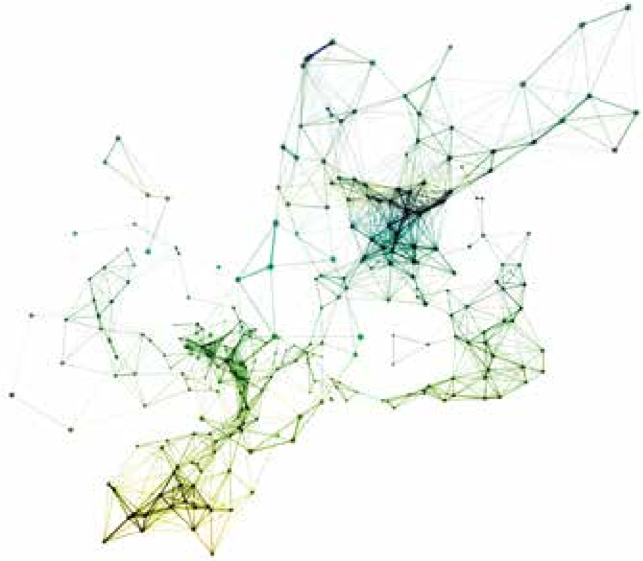
Most installations nowadays consist of standard Linux clusters, x86-based systems with an interconnect, mostly InfiniBand, using some standard Linux distribution. These systems serve very well a huge number of scientists and engineers, and depending on the configuration mostly provide the scalability to also tackle challenging scientific problems.

So clearly these systems represent the "standard" for most users. And they represent the basis for the first experiences of most students, and thus for the future of code developments.

In some cases specific requirements show up, and are represented in the requirements of the site, for example:

- ➔ A need for a big SMP-type system with large memory because of a code which is parallelized on shared memory, but not for distributed memory.
- ➔ Some kind of special I/O-features, e.g. for frequent small-block I/O, often for example in chemistry.
- ➔ In rare cases a need for a GPGPU to accelerate special computations.

All these special cases normally lead to minor extensions of an otherwise architecturally homogeneous configuration.



But there are also other cases, and historically these first showed up at some large centers and in the meteo-community. In most cases the workload consists of different types of applications, and the idea is to provide the best or at least a very reasonable platform for each important application.

As one customer from a renowned institute for oceanography once put it: "I want the optimal hardware for each application of interest". In this case there were a handful of applications with quite different characteristics, some matching a scalar architecture, others needed a vector system.

Similarly the need for various systems is easy to understand in case of weatherforecasting: Initially there are tremendous amounts of data from observations from satellites, ships, air planes, weather stations, which need to be assembled and put into some consistent format. In addition such data are archived and partially put into huge databases. Then these data are filtered, preprocessed, sorted, reformatted etc. From there a global weather model is used to calculate long-term and large-scale weather forecast, in most cases on a global scale. Subsequently the output of such forecasts is refined for specified regions, which are of interest to the institution running this system. This leads to "regional forecasts". And in some cases these are further refined, e.g. for very precise short-range forecasts for airports. Then results have to be analyzed, interpreted and visualized. Moreover they are archived together with the initial data.

Clearly this describes a complex workflow, part of which can consist of several tasks running in parallel, with or without human intervention, other parts, especially the weather models, are clearly HPC-type applications, in fact the most demanding ones and historically often the first to use new programming paradigms, because of the numerical challenges.

So in both cases it is understandable that sites will increasingly run different architectures to support the variety of applications. And such strategies have been followed already since the mid-90s by some NEC customers, and still today this is the case.

NEC's contribution: a total solution offering

NEC has quite some experiences with regard to hybrid configurations since, for example, the DKRZ was hosting the biggest scientific database in 2005, used for climate data. This database was provided by NEC, as a "side effect" to an HPC-project, which led to a large vector installation. And this database was strongly integrated into the whole workflow of climate-simulations.

Similarly NEC serviced quite some weather sites in different countries with highly complicated configurations, including 24/7 support for the operations.

Coarse-granular hybrid configurations

This fact is reflected in the hardware. NEC will provide a Linux-based complete cluster at a competitive price, a variety of filesystem appliances for different application needs, GPGPUs and NVME if needed, plus, and that's the difference, a tightly integrated vector system to tackle the most bandwidth demanding applications. It will even feature "hybrid computing" by the provision of an MPI-implementation for the combined use of scalar and vector nodes in one application.

A simple case of such a hybrid application would consist of a weather forecast application running on the vector nodes with I/O-servers, independent MPI-processes to asynchronously handle the tremendous I/O, running on scalar nodes connected to a fast parallel filesystem.

One could also imagine very challenging cases of hybrid applications distributed over vector and scalar nodes, for example climatology simulations. These normally consist of an ocean-code, a model for the dynamics of the atmosphere, atmospheric chemistry, and some other ingredients like sea ice simulations, models of the soil and much more. All these components are coupled by one of the known couplers, OASIS or PRISM or the like.

Atmospheric dynamics and ocean-code are naturally very bandwidth-limited, these should be handled by the vector architecture, while atmospheric chemistry, consisting of a simulation of reaction kinetics running relatively decoupled on each individual grid point, is mostly coded in a very scalar, but also well scaling way. So these portions could well be handled by a scalar cluster.

Consequently the ideal system to execute such simulations is a combination of both components, vector and scalar nodes.

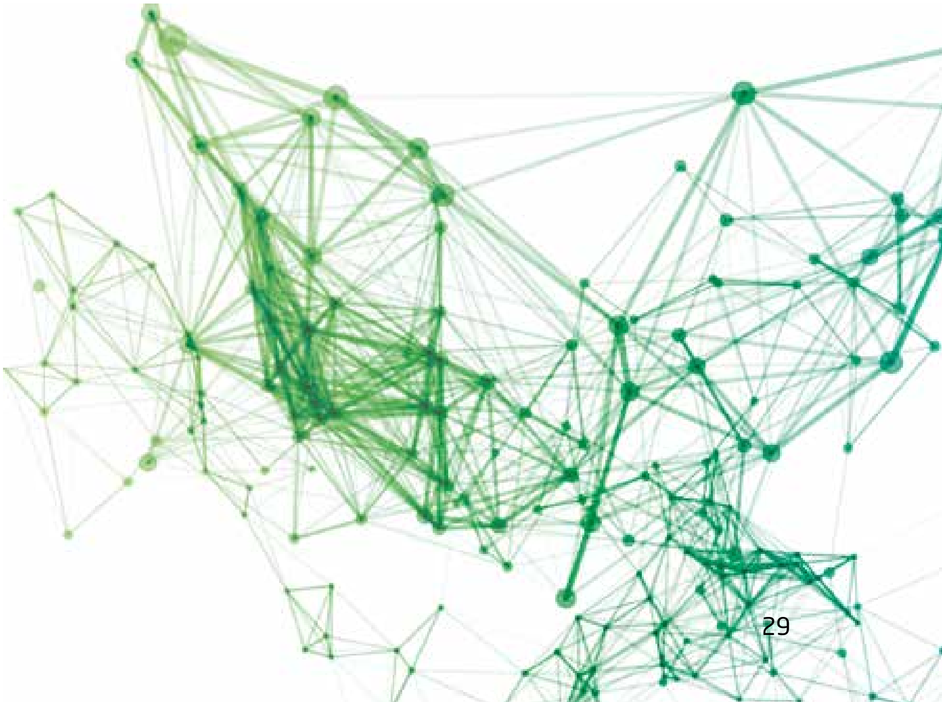
There are other situations, which could be well handled by a hybrid configuration. For example it would be natural to execute very complex simulations on a vector system while running the visualization necessary to drive a cave would be processed by scalar nodes which get the data via MPI. Or one could think about a coupled simulation of the dynamics of an air foil, perhaps simulated by ABAQUS, coupled with a compressible airflow simulation running on a vector.

So in conclusion it starts by providing the best hardware for each type of application in a throughput environment, hybrid workload, over a workchain of applications which represent a production process, and for which individual steps of the process logically depend on each other in some defined way, a hybrid workflow, up to the challenging simulations consisting of coupled components.

The Future

The time is now for some breakthrough in the HPC-market. NEC's legacy in vector computing has been successful in the past, but then again the technological issues that all architectures are facing today are addressed by a vector architecture in a natural way.

And the tight integration of the vector nodes into an otherwise "standard" environment opens a whole new area of supercomputing, namely hybrid applications. This enables the utilization of the most efficient hardware, especially also with regard to energy efficiency, which is the dominant topic in HPC today.



Vector Hardware

Roadmap

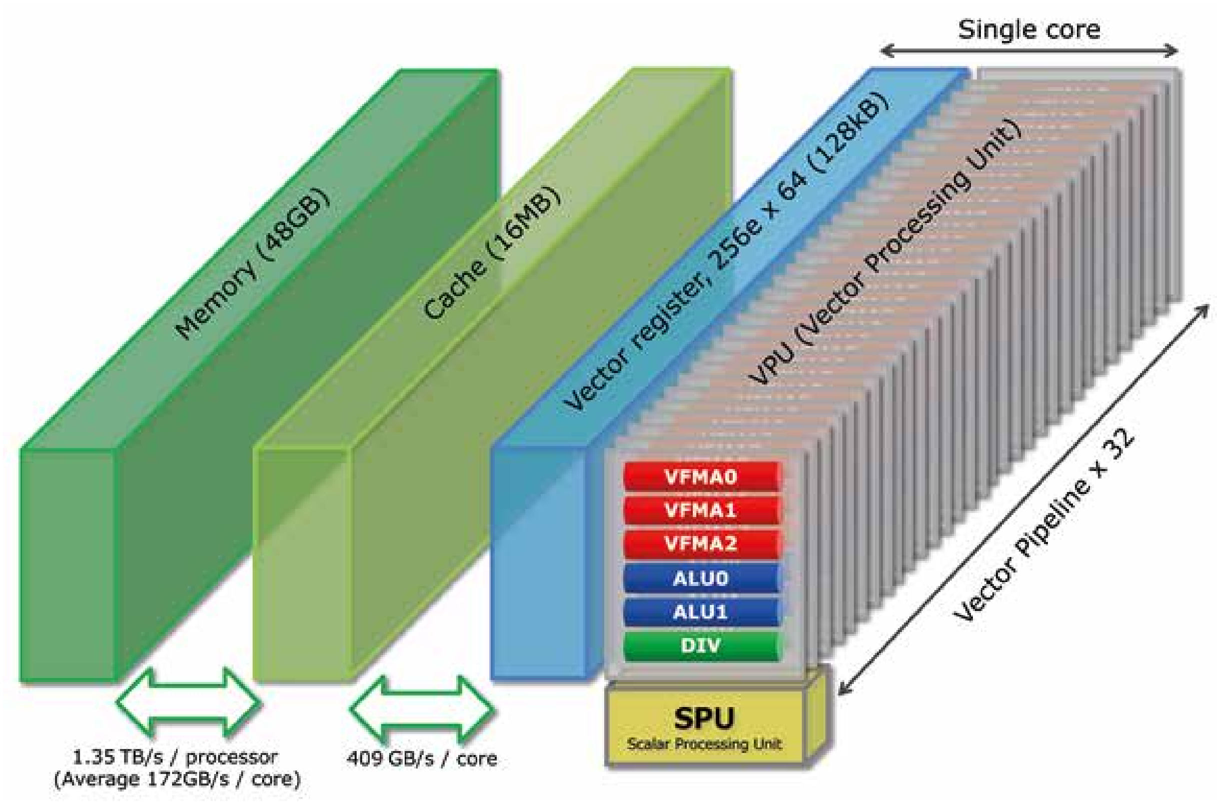
NEC continuously improve the vector architecture in order to provide higher sustained performance. Evolutions of SX-Aurora TSUBASA are shown in this subsection as a first step.

The launch of SX-Aurora TSUBASA was in 2018. This is the first generation of the SX-Aurora TSUBASA series with VE type 10A/10B/10C called a VE10 generation. The processor of the VE10 generation had the world largest memory bandwidth, 1.22TB/s. This high memory bandwidth contributes to provide higher sustained performance especially for memory bandwidth intensive applications.
As an enhanced version of the VE10, NEC introduces a VE10E generation consists of VE Type 10AE/10BE/10CE from beginning of 2020. The new generation pursues higher sustained performance with higher memory bandwidth, 1.35 TB/s per processor.
A VE20 generation is going to appear in 2020 as the successor of the VE10E generation. The memory bandwidth per processor is going to be higher with higher calculation capability.

After the VE20 generation, a VE30 generation having a newly designed vector processor is planned. Through these VE generations, each processor provides the highest level memory bandwidth to accelerate memory bandwidth demanding applications with high power efficiencies. In this section, details of the VE10E generation are described.

Vector Architecture

As stated the SX-Aurora TSUBASA features a vector architecture. So there are vector registers and vector pipes in the first place.



Vector Core

The CPU of SX-Aurora TSUBASA has eight independent vector cores.

Vector Registers

There are 64 fully functional vector registers per core, which can feed the functional units or which receive results from those.

This is the first significant and important enhancement compared to the previous SX-systems, which only had eight vector registers, plus 64 so-called vector-data-registers used for keeping intermediate results on the CPU, but lacking the full functionality of a vector register, for example to act as an input to the functional units.

The compiler has a lot more options now to optimize the register usage and consequently this will enhance the efficiency of the system.

The vector registers have 256 entries of 8-Byte-width, thus being able to handle double precision data at full speed. So the full set of registers represents 128 kByte of data readily available for computations at the speed of a register!

Vector Pipes, or Functional Units

The single core of the SX-Aurora TSUBASA processor has three FMA-units, plus one for divide and sqrt, per core. FMA stands for "Fused Multiply-Add", and reflects an expression like

$$a * b + c$$

The importance of the pipe for divide and sqrt is often underestimated. Indeed such operations are not dominating in normal scientific applications. But once they are needed, they often pop up to be limiting, or at least contribute significantly to the execution time. For example this often happens in the case of molecular dynamics, when distances are important for the calculation of forces between atoms. Other architectures emulate such operations by a software approximation, which can be time consuming.

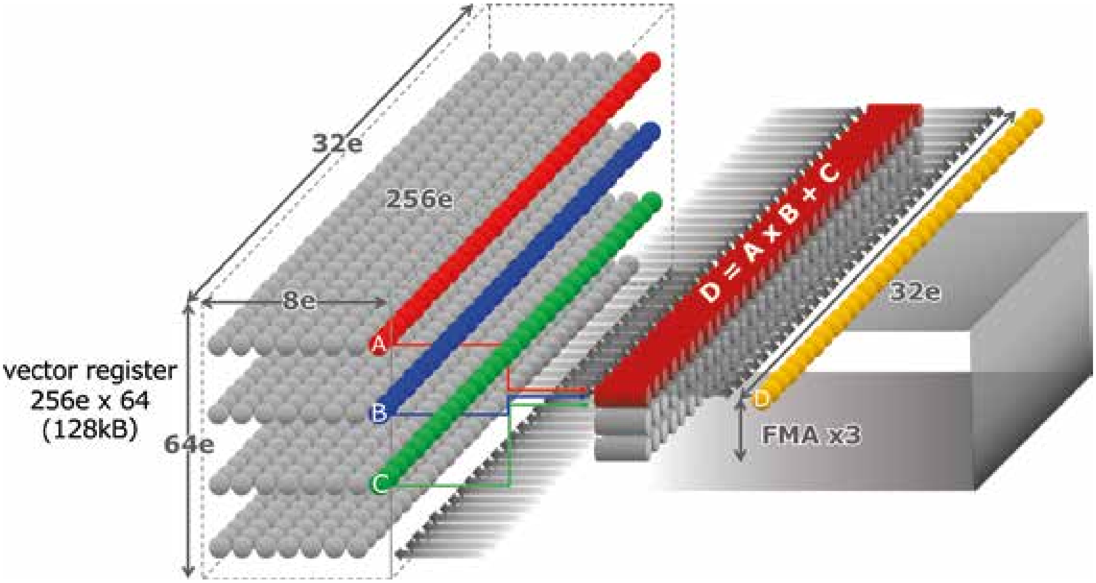
In order to provide the high computing capability those pipes are inherently a 32-way parallel, means, each FMA unit produces 32 results per cycle.

Traditionally only add- and multiply-operations are counted when computing the peak performance of an architecture. For the system this is then

3 * 2 * 32 = 192 floating point operations per cycle

There are more pipes for fixed-point arithmetic and logical operations. And all together they are fully connected to the vector registers by a fully non-blocking crossbar.

A very schematic view on this architecture:



An important enhancement of the system compared to the previous SX-architecture lies in the support of 4-Byte-arithmetic at twice the speed. This means for single precision the system can do 384 floating point operations per cycle. For some application fields, for example crash simulation or seismic imaging, this is considered sufficient precision for certain cases.

Core Frequency

The power-consumption of CPUs is an important problem nowadays, not only for cooling problems, but also for the calculation of the total cost of ownership. Power consumption scales roughly like frequency cubed. At the same time it is inefficient to have a high frequency for the computation while the memory bandwidth, which is dictated by other technologies, cannot deliver sufficient data to justify such a frequency. To achieve such a balance there are two different versions of CPUs with frequencies of

1.4 GHz and 1. 6 GHz

Consequently the single-core performance computations are

Double precision: 268.8 GFlops and 307.2 GFlops

Single precision: 537.6 GFlops and 614.4 GFlops

To put this into perspective, one core of the system is about as fast as one CPU of the previous product SX-ACE!

Scalar Operations

The scalar portion of the core is not just a scalar CPU, it controls all available resources. The scalar CPU has complete information about all ongoing activities, it can schedule functional units and registers, both vector and scalar registers, and can even reorder operations on several levels to enhance efficiency. This is highly important as it serves to hide latencies, also latencies from memory access, and this was always a big contribution to the efficiency that NEC's vector machines could realize.

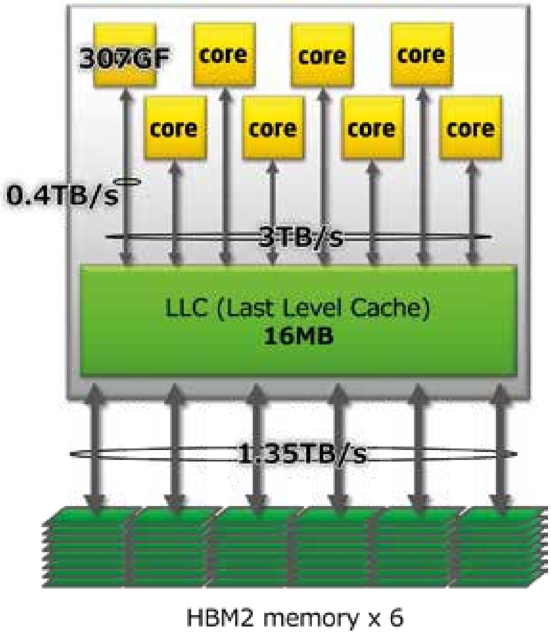
This is important to understand, even the best vector code comes with a lot of scalar operations, like updating address registers for vector loads, incrementing and checking loop counters etc. Such operations need to be conducted while the vector pipes are active. But there is plenty of time, or cycles, to complete these operations, which therefore do not affect the performance of the actual computation in the vector pipes.

As a matter of fact already on previous generations of NEC's vector systems optimizing a code caused the hardware counter for "Instructions per second" to decrease, while the counter for "operations per second" was increasing. A vector instruction implies 256 operations in case of an add or a multiply operation for double precision, and once the vector pipes are fully active and no other instructions are necessary any more, instruction issue is reduced. The scientific user needs operations, not instructions.

The scalar portion of the CPU has two level 1 caches for instructions and for operations, each 32 kByte. As an enhancement compared to the previous architecture the system features an additional level 2 cache of 256 kByte, which further enhances the scalar performance and especially decrease the time spent on instruction fetching.

CPU

The following figure shows a block diagram of the vector CPU and connected memories.



Before this description, the following sentence should be added.

This CPU mainly consists of eight vector cores and LLC (Last Level Cache) shared by the eight cores.

Because this CPU runs at 1.4 GHz or 1.6 GHz, the CPU performances in double precision and single precision are:

"Last-Level-Cache" (LLC)

Already with the SX-9 and the SX-ACE NEC introduced the first ever "vector cache", called "Assignable Data Buffer" (ADB) at that time, because it could be software-controlled, data to be cached were "assigned" to reside there. These ADBs were not shared between CPUs or cores.

For the system NEC developed a shared "Last-Level-Cache" (LLC), the first shared vector cache ever. This shared LLC serves all cores together.

The LLC has a "write back" policy, which means data coherency between different cores, LLC and memory is always easily ensured.

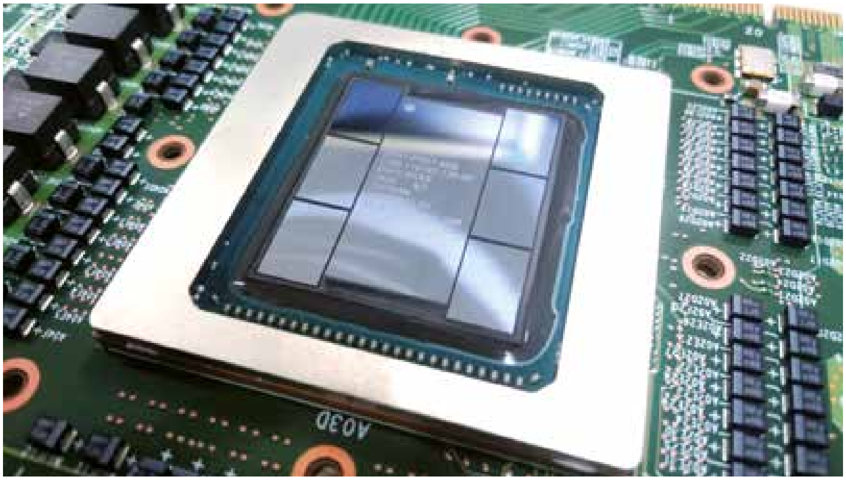
At the same time this kind of architecture lends itself easily to a shared memory parallelization, by autoparallelization or OpenMP, while MPI would be used to parallelize over different CPUs.

The last level cache has a line-size of 128 Bytes, and some additional features are implemented to increase the efficiency for strided stores or scatter operations.

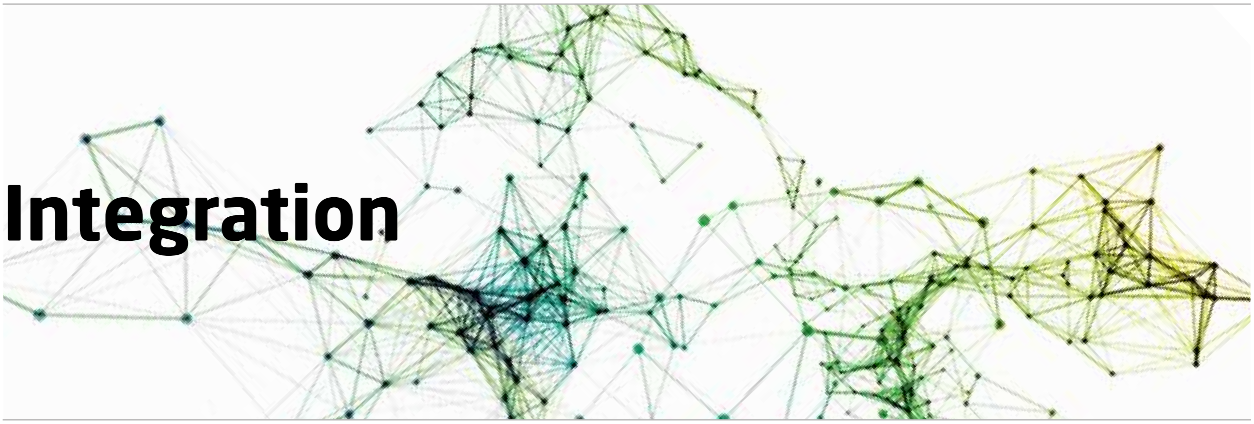
Memory

NEC has investigated the existing memory technologies available for a high-end system, and has chosen to utilize the second generation of the so called "High Bandwidth Memory" standard, HBM2. Such a component is realized by stacking four or eight dies together, and it achieves 230.4 GB/s bandwidth while providing four or eight GBytes of capacity.

Six of these components and the CPU are connected by a so-called “Silicon interposer”, a kind of die to mount on and that way connect all pieces as the following picture shows.



NEC uses six such HBM2 components for a total of 24 GByte to 48 GByte per CPU, and 1.35 TB/s bandwidth, which is industry leading, the best bandwidth in the market at this point in time.



Integration

Basic concept

The idea is to take the best of both worlds, the “standard x86-Linux-cluster” and the “high-end vector architecture” and merge this into one product. Combining the favourable features in particular with regard to operating system and user environment the system reflects the basic design strategy sketched earlier.

Realization

The standard environment is provided by a normal x86 cluster node, it is called “Vector Host” (VH), the vector architecture is added as the so-called “Vector Engine” (VE). These two components are connected by a PCIe Gen.3 x16 interface. The support for forthcoming standards will not pose difficulties.

VE has been realized by PCIe card implementation. One vector CPU, the main memory of the six HBM2 implementation are mainly implemented on the card. A PCIe connector to connect VH is PCIe Gen.3 x16. An actual power consumption of this card is around 250 W, but it strongly depends on each application characteristics executed on the card. A memory bandwidth bound application requires much less power consumption per card because an activation rate of each arithmetic unit becomes lower.

The VE card has three variations called VE1.0 Type 10AE, Type 10BE, and Type 10CE. Type 10AE is a high end specification, and Type 10CE is an entry version.

VE Type	frequency (GHz)	core	processor			
					memory	
		GF	cores	DPTF	BW	size
					TB/s	GB
Type 10AE	1.6	307	8	2.45	1.35	48
Type 10BE	1.4	269		2.15		
Type 10CE					1.22	24

Supercomputer with Frontend

From the type of connection one might be tempted to compare the product with a GPGPU. But it is important to understand,

“SX-Aurora TSUBASA is not “yet another accelerator”

The principle of operation is drastically different, in a way the opposite of how typically accelerators work. On NEC’s system the complete application runs on the vector CPU, not just parts which are offloaded in some way using paradigms like CUDA or OpenACC or the like.

Rather think of the vector CPU being the major workhorse, the VE, and the x86 node, the VH, being kind of a frontend, which handle compilation or take over typical OS-duties, like resource scheduling, or some specific workloads, or like I/O requests, interactions with filesystems for which the vector CPU is not ideal and much to precious anyway.

The typical usage of accelerators or GPGPUs implies to start on the x86 host and then offload pieces of the workload, like a complete loop nest, to the GPU. This implies to transfer a lot of data back and forth, and in addition this should happen frequently, at the granularity of loop nests, because otherwise the utilization of the GPGPU would be low.

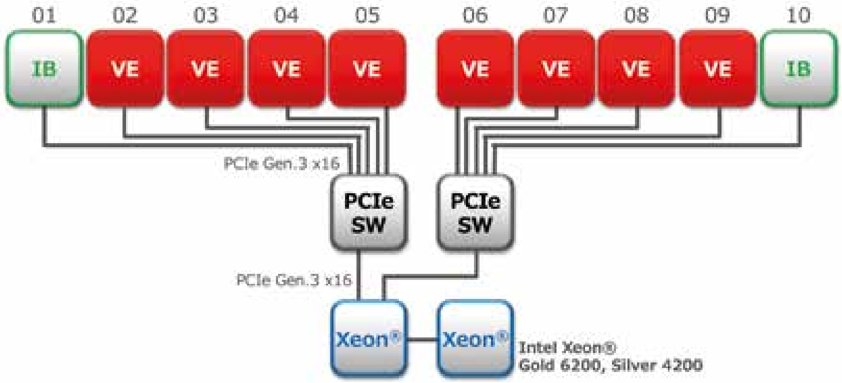
On the NEC system it is the other way round, only those pieces of the application are offloaded to the x86 system which imply activities for which the vector architecture is not ideal, for example I/O requests. In normal applications these should occur less frequently and in addition the amount of data to be transferred should normally be smaller.

Interconnect

Initially Mellanox InfiniBand EDR and HDR are used.

Note that the vector nodes can communicate with any other vector node in the fabric, and also with any other scalar node, without involvement of the vector host.

The basic configuration for high-end systems consists of eight VE cards, two Xeon processors, two PCIe switches, and two InfiniBand HCAs as a one server module.



The VEs are connected to the HCAs and to the Xeon CPUs of the VH by the PCIe switches.

MPI-messaging will be implemented by NEC's own MPI, which is "hybrid", means it allows to utilize scalar and VE cores in one application.

Naturally VEs need to exchange information with each other. If they are connected to the same PCIe-switch this kind of messaging is "direct", does not involve the VH except for the initialization mpi_init.

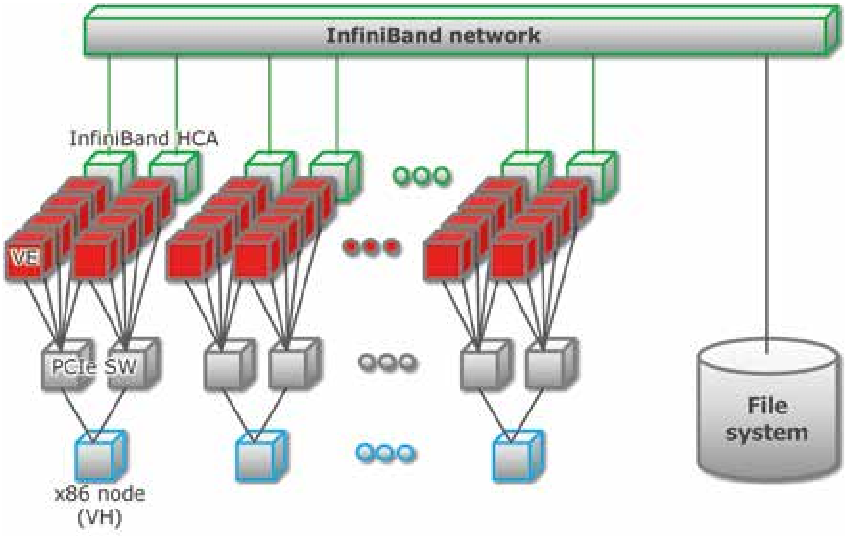
If the VEs belong to the same VH, but to different PCIe switches, the messages will be routed via the PCIe switches through the PCIe root complex of the Xeon.

Similarly the VEs can exchange data with the VH directly through the PCIe switch, a hybrid utilization.

And moreover the VEs can communicate with other VEs or other scalar cluster-nodes via the IB fabric, which is connected to the PCIe switches.

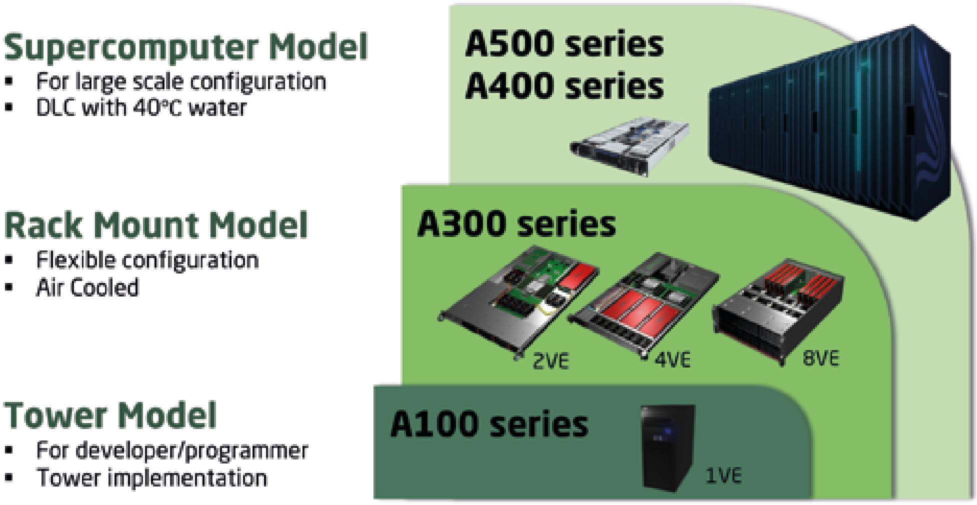
Overall this concept allows for a very high flexibility both with regard to configurations as well as with regard to its usage.

The following figure shows an example of large scale configuration. VE cards are connected with the InfiniBand network as a large scale vector supercomputer.



Products of SX-Aurora TSUBASA

SX-Aurora TSUBASA has mainly three variations of product, A100, A300, and A500 as following figure shows. Such the variety of the product line up can cover wider range of the market or use cases. Each variation of SX-Aurora TSUBASA has each characteristics as following figure shows.



Supercomputer Model

- For large scale configuration
- DLC with 40°C water

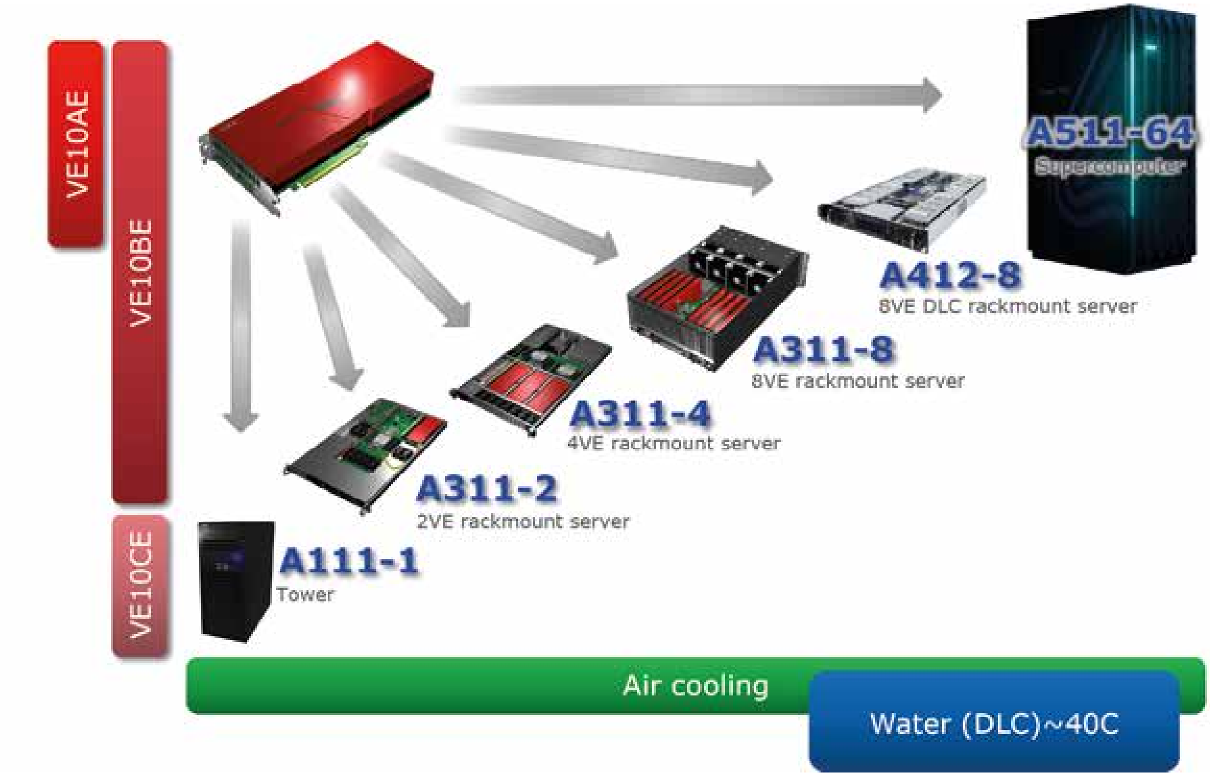
Rack Mount Model

- Flexible configuration
- Air Cooled

Tower Model

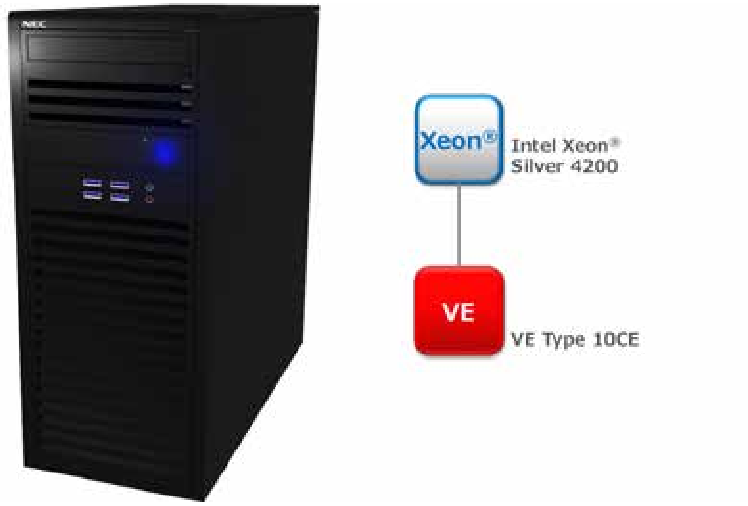
- For developer/programmer
- Tower implementation

The horizontal-axis shows the cooling mather of each product, which are an air cooling, water cooling door, or a direct liquid cooling (DLC) with water flow. The vertical-axis shows supported VE versions, Type 10AE, Type 10BE, and Type 10CE.



The A100 series is a tower model, which can be used on a desktop. This model is developed to cover kind of personal use for developers and programmers, and mainly consists of one Xeon processor and one VE card of Type 10CE.

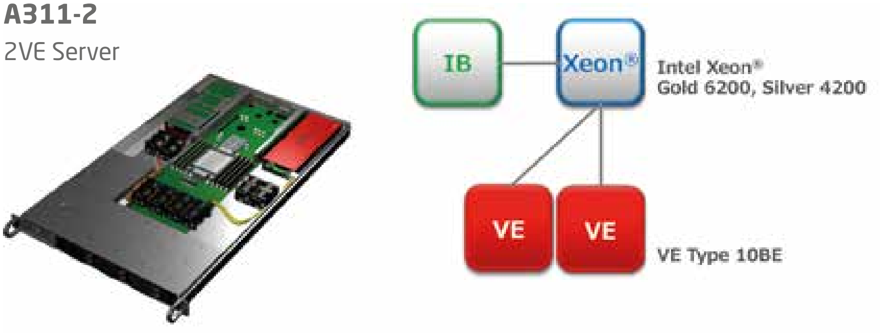
A111-1
1VE Tower



The A300 series is a standard rack mountable model with air cooling. There are three types of products in this series, A300-2, A300-4, and A300-8. The numbers following A300 are the number of maximum VE cards per each A300 product. Due to the standard rack mount implementation and the air cooling, this series has high configuration flexibility as same as a de-facto standard x86 servers. The supported VE cards are Type 10B and Type 10C.

A block diagram of A100-2 is shown in the following figure. This model consists of one Xeon processor, up to two VE cards, and one IB HCA with 1U implementation.

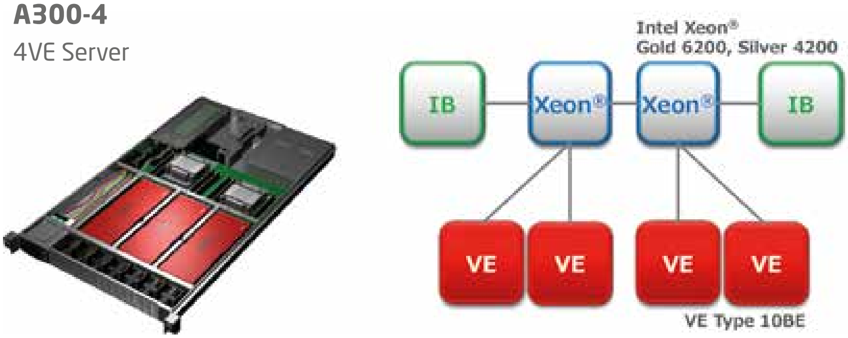
A311-2
2VE Server



A300-4 can be implemented up to four VE cards per server. The server mainly consists of two Xeon processors and up to four VE cards of Type 10B or Type 10C. Moreover, up to two IB HCAs can be equipped with this server.

A300-4

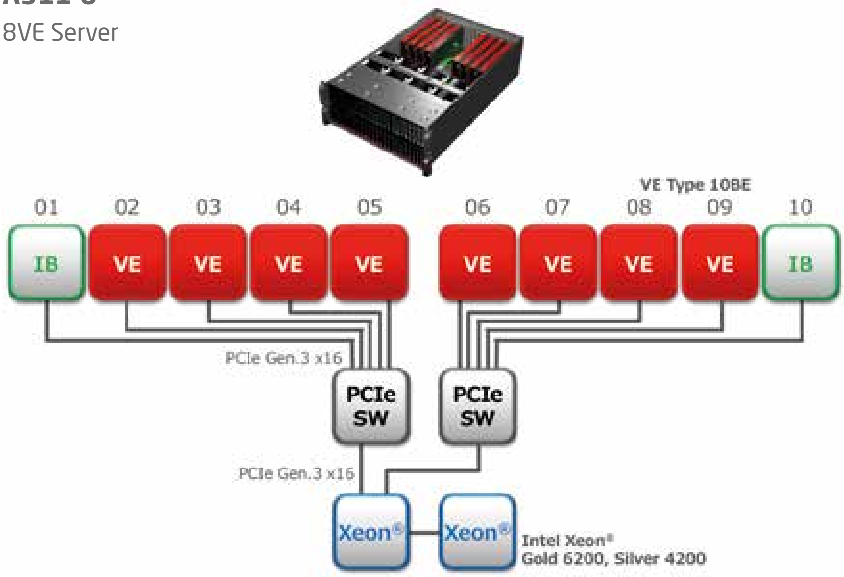
4VE Server



High vector processing power and high memory bandwidth are provided by the A300-8 series. This sever is designed for large scale vector processing by implementing up to eight VE card per two Xeon processors. A big differentiation from A300-2 and A300-4 is that two PCIe switches are located between VE cards and Xeon processors. Due to this implementation, such the eight VE cards can be connected to one Xeon node, and direct MPI communications between VE/VE or VE/IB HCA via a PCIe switch are provided. VE card Type 10B and Type 10C are also supported by this product.

A311-8

8VE Server



The high end product of SX-Aurora TSUBASA is A500-64. Up to 64 VE cards are implemented into one dedicated DLC rack.

Basically, the server unit of A300-8 is used with implementing DLC units and optimizing full rack cooling for a large scale supercomputer operation. Eight servers are implemented into the rack. The high end VE card of Type 10AE is only supported by this DLC supercomputer. The Type 10B VE card is also supported. In order to reduce cooling cost, an inlet water temperature to cool this rack is allowed up to 40 degrees °C called a hot water cooling.

Coding, a User's Perspective

What does this now mean for a user, what characteristics of an implementation is required to utilize such an architecture?

Since the length of the vector register is 256, loops are strip mined into pieces of this size. Vector instructions only make sense when the loop length is appropriate, not necessarily 256, in principle the vector pipes can be used with any trip count, but a loop-length 8 or 16 would not lead to the desired performance.

From the mathematical formulation of typical scientific, based on a grid, or a huge number of particles, finite elements or volumes, there is always a sufficient amount of data parallelism in codes, just because it results from the underlying characteristics of the problem and how this is expressed mathematically, for example as a partial differential equation. Nature is local, means things can be described by partial differential equations in some way, and thus are inherently parallel and providing the notion of "neighbourhood". The application should be written in such a way that the compiler could clearly identify such underlying structures.

Surprisingly, this mathematical or physical fact is often not reflected in the way the applications are written, sometimes the structure is rather hidden, or at least not well reflected in how the variables are organized in memory.

People often think in a certain way when coding:

I have an element, a grid point, a particle, an equation, whatsoever. What am I going to do with it?

The vector paradigm is simple, and in a way just the opposite:

I have to execute a certain operation. On which elements, grid points, particles, equations, can I apply it simultaneously?

If this is your way of thinking, you will inevitably write vector code.

What does it imply? For example, a grid point does not have a mathematical meaning, the complete grid is the abstraction of space or space time based on which the mathematical problem is formulated. Similarly, a physical field of whatever kind should be considered a mathematical entity, not the individual value of the density as a "variable" living on a grid point.

Consequently, the grid and the fields living on it should be "recognizable" in the data-structures.

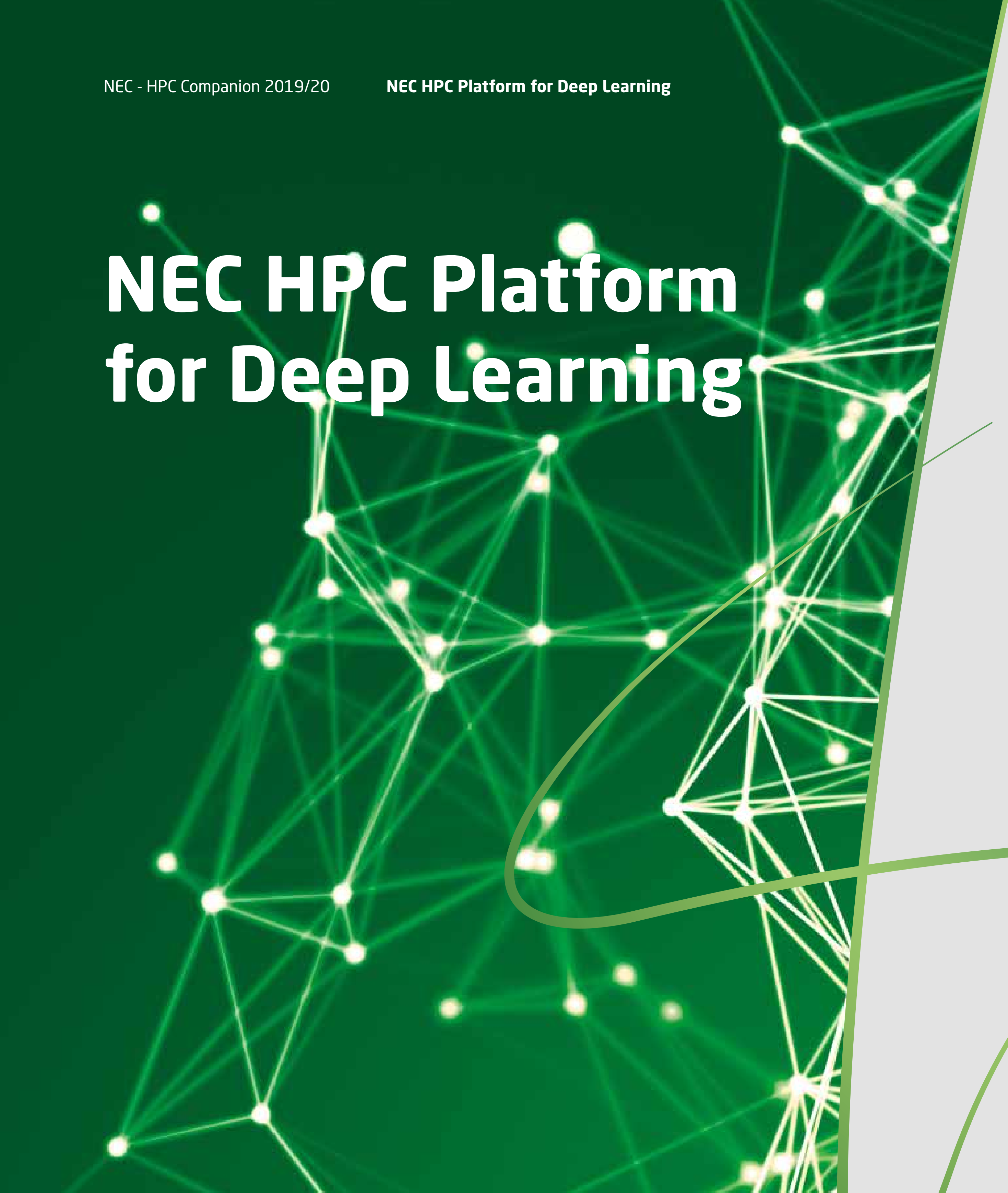
There are codes which use a C structure to represent a grid-point and the "variables", density, temperature, whatsoever, and in the worst case these grid points are connected as a linked list. No compiler on earth can identify the hidden data parallelism any more. And in such a case it is close to impossible to ensure cache locality, and to hide the latency of memory accesses. But both code features are necessary to effectively utilize any modern hardware.

Instead, the fields should lead to arrays, living on a grid. This is simple in the case of structured grids, not so simple but feasible on unstructured grids, admittedly difficult in the case of adaptive refinements for enhanced resolution. Admittedly, there are codes, which do not have anything like a grid. But mostly similar ideas can be applied.

It is a known fact since many years, and although initially it was mainly a problem of vector architectures, nowadays it is more general because of the need for data parallelism to utilize SIMD instructions: in C one should organize variables as a structure of arrays, not as an array of structures. Similarly, more modern: the grid and the mathematical fields should be objects, not the combination of grid point, which is a collection of coordinates, plus the variables on it.

It is basically also this understanding which some years ago lead to the idea of "domain specific languages" (DLC), namely to define a tool to describe an action to be applied to a whole field, for example the application of a differential operator. In the ideal case vendors would provide the means to translate these constructs somehow into machine code in an optimal way for a given architecture. In any case and obviously, this direction exactly represents the vector paradigm.

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. Complex problems, such as object detection, text translation and speech recognition, can now be solved with relatively small effort leveraging Deep Learning techniques.

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.

NEC HPC platform for Deep Learning

A Brief Introduction to Deep Learning

At the core of Deep Learning there is a set of algorithms generally called artificial neural networks. An artificial neural network is composed by a set of atomic units, called neurons, which perform very simple computations. When properly combined together, the simple computations of the neurons are able to implement much more complex functions, e.g., to describe the content of an image. The combination of the different neurons is performed using an automated training process, which leverages the *backpropagation* algorithm. When provided with data, the combined computation performed by the neurons is learned via backpropagation by adjusting weights values associated with each neuron, thereby making a neuron relatively more or less important for the target mathematical function. Generally, a neural network requires a big amount of data to be trained, and the more data is provided to the training process, the closer to the target function is the neural network's learned function.

Usually, neurons are grouped together in layers that perform different conceptual operations, depending on the neurons' types and their interconnections. These layers are used as basic building blocks for neural network architectures and, as such, a neural network is generally described as a sequence of layers.

There are three main broad types of neural network *architecture*s that have been widely used in the last few years:

- Multilayer perceptron (MLP)
- Convolutional neural networks (CNN)
- Recurrent neural networks (RNN)

While these neural network architectures share many common types of layers, they generally provide better results for different specific tasks. For instance, MLP are a good fit to perform tasks on tabular data, convolutional neural networks (CNN) are capable of dealing with data that show some spatial correlation, such as images, and recurrent neural networks are generally well-suited to work with time series.

Example Applications

While use cases such as image recognition are already a quite known field of application for neural networks, there are many other newer fields where this machine learning tool has recently provided interesting results. For instance, NEC is exploring the application of deep learning to several digital health applications, which we briefly describe next.

Digital pathology. Pathologists aim to diagnose a patient's current health status as well as forecast how the patient's status is likely to progress in the future. High-resolution images of cells and tissues from sophisticated microscopes are one of their most important tools. Robust detection, segmentation, and classification of tumor, normal tissue, and respective cells are an integral challenge for diagnosis and prognosis from pathology stains. At NEC, we have developed novel CNN architectures to solve problems like characterizing tumor morphology and counting immune cells as diagnostic and prognostic biomarkers.

Immuno-oncology. Immunotherapy is a promising therapeutic approach for combating cancer by teaching the patient's own immune system to recognize and destroy cancerous cells. However, each patient is unique, and precision medicine approaches are needed to give each patient the best chance for a positive outcome. We are using novel graph neural networks (a particular sub-family of CNNs) and RNNs to improve the efficacy of these treatments by confronting challenges like genomic characterization of cell subpopulations, prediction of adverse events following checkpoint inhibitor treatment, and selection of neoantigens as vaccine targets. This technology has been approved by the FDA and EMA for use in clinical trials in the US and Europe.

Electronic health records. Modern hospital information systems keep very detailed patient records. The records include many data modalities, including patient demographics, lab measurements, notes from physicians and nurses, movement data during hospital visits, and more. We are applying graph neural networks to extract valuable patterns from these records including outbreaks of dangerous pathogens like Candida auris or methicillin-resistant Staphylococcus aureus (MRSA) within hospitals and prognostic risk indicators for both short- and long-term patient outcomes.

Drug discovery and repurposing. Pharmaceutical research aims to discover high-quality drugs which are effective, safe, and cost efficient. Many diverse factors affect these qualities, such as the chemical properties of the drug, how it interacts with a disease on the molecular level, and the genetic background of selected patient cohorts. We represent these myriad properties using a biomedical knowledge graph and use graph neural networks to identify novel relationships like side effects from untested drug combinations and genes affected by diseases. These relationships increase the likelihood that drugs selected for further investigation have a high chance of success; further, the discovered relationships uncover opportunities to repurpose existing drugs for new diseases.

Software for Deep Learning

The quick evolution of Deep Learning has been propelled by the availability of a number of software frameworks that simplify the development of neural network algorithms. 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 layers. Writing a new neural network is as simple as sticking together a few lines of code using the python programming language.

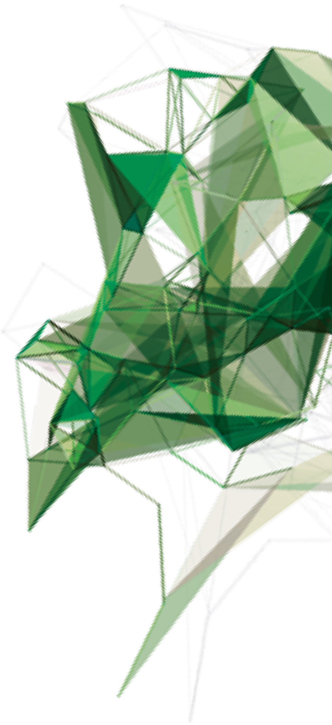
For instance, a common code snippet could look like the following:

```
import framework
model = framework.ModelZoo.init(“MyNeuralNetwork”, …)
input = (…load input…)
result = model(input)
```

The frameworks will then take care of all the operations required to enable the training of the neural network and its execution.

Under the hood, each of these frameworks maps the high-level API to low-level optimized hardware specific computation libraries. For example, the cuDNN library is used with NVIDIA GPUs that use the CUDA computation model. Given the high-level API, the user of the framework can ignore the low-level hardware details, focusing on the neural network algorithm design. This also provides hardware agnostic implementation of the neural network, since the hardware executor, e.g., a CPU or a GPU, is managed by the framework transparently.

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. In some cases, the execution of different layers could be instead merged together from a computational perspective, optimizing the use of the underlying hardware, e.g., avoiding the data transfer that usually happens when starting and finishing the computation of a layer.



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 analyzes the neural network descriptions to provide additional hardware compatibility and improve performance.

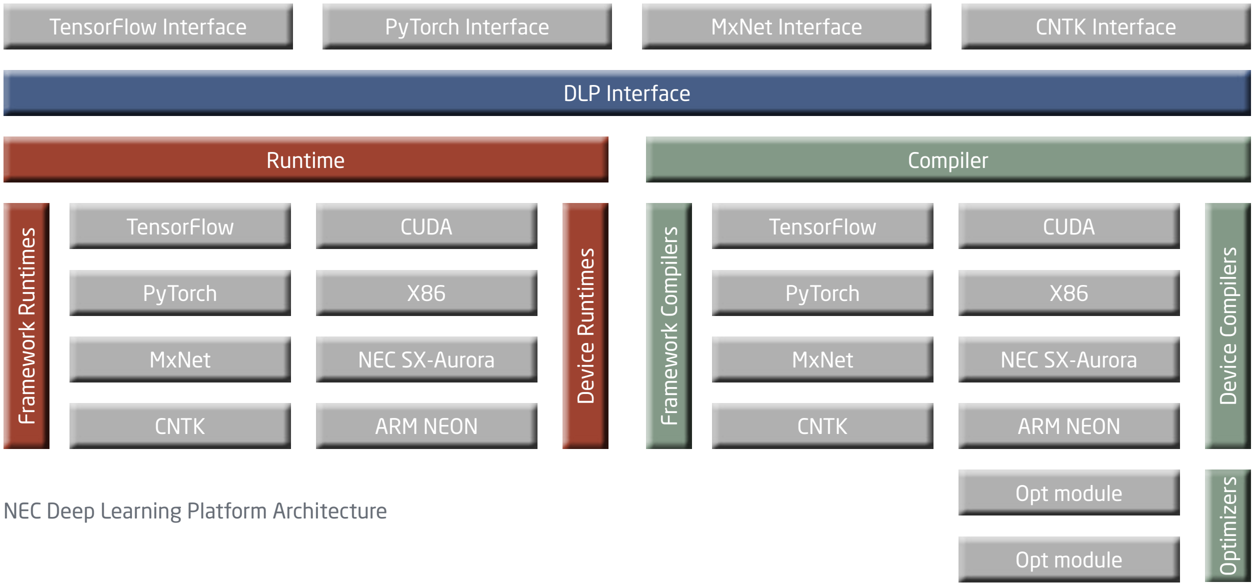
The NEC DLP reads the description of a neural network defined in the framework's syntax, analyses its structure, and automatically derives a set of optimized computations for the specific neural network the user has described. Then, in addition to the low-level computation functions available in libraries such as NVIDIA cuDNN or Intel MKL-DNN, the NEC platform also generates any additional low-level computation function that may be required by the optimized neural network, for a target hardware executor. 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 fully functionally equivalent for the user, which can perform 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)
```

Two lines of code are added to the previous example to load the NEC DLP and to apply its optimizations to the neural network model.



NEC Deep Learning Platform Architecture

From an architectural perspective, NEC DLP is shown in the figure above. The rest of this section provides more details on the different architectural parts and on the supported features.

Optimization techniques

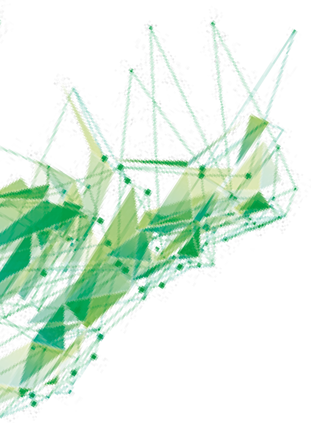
Deep Learning frameworks, like TensorFlow and PyTorch, execute their neural networks on a layer-by-layer basis, which can result in an inefficient use of the hardware executor's memory hierarchy and thereby in long execution times. NEC DLP addresses the problem by generating a new custom computation 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 transformations 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 patterns that can modify the network's structure, improve the reuse of data buffers, optimize the use of the device's memory hierarchy and computation units.

Software compatibility

The speed at which deep learning architectures are developed, tested, and open-sourced is staggering. For instance, almost every other week a pre-trained model for natural language processing is published (such as ELMO, BERT, XLNet) and shown to be superior to previous ones on established benchmarks. In many cases, these models are written in one specific deep learning framework, optimized for efficiency, and pre-trained on very large datasets. To allow businesses to utilize these models is straight-forward if the deep learning framework matches that of the existing development environment. In many cases, however, there is a mismatch and it is time consuming and expensive to port new architectures and layers to another framework. It is time consuming because the implementations are highly optimized and, therefore, non-trivial to port to other deep learning frameworks. It is expensive because pre-training a model often takes several days on specialized hardware.

NEC DLP, working as a middle layer between specialized hardware and deep learning frameworks, allows one to design and experiment with deep learning models with cross-framework layers. For instance, we can use NEC DLP in Tensorflow to include PyTorch layers and vice versa.

For example, TensorFlow does not support the ConstantPadding layer, which is supported in PyTorch instead. With NEC DLP, the user can use “dlp.nn.padding(type=dlp.PaddingType.Constant, value=N)” to use these layers across frameworks. As the dlp.nn API is identical in all frameworks, a model can also be written entirely in this API and be directly interchanged between the frameworks.



Hardware support

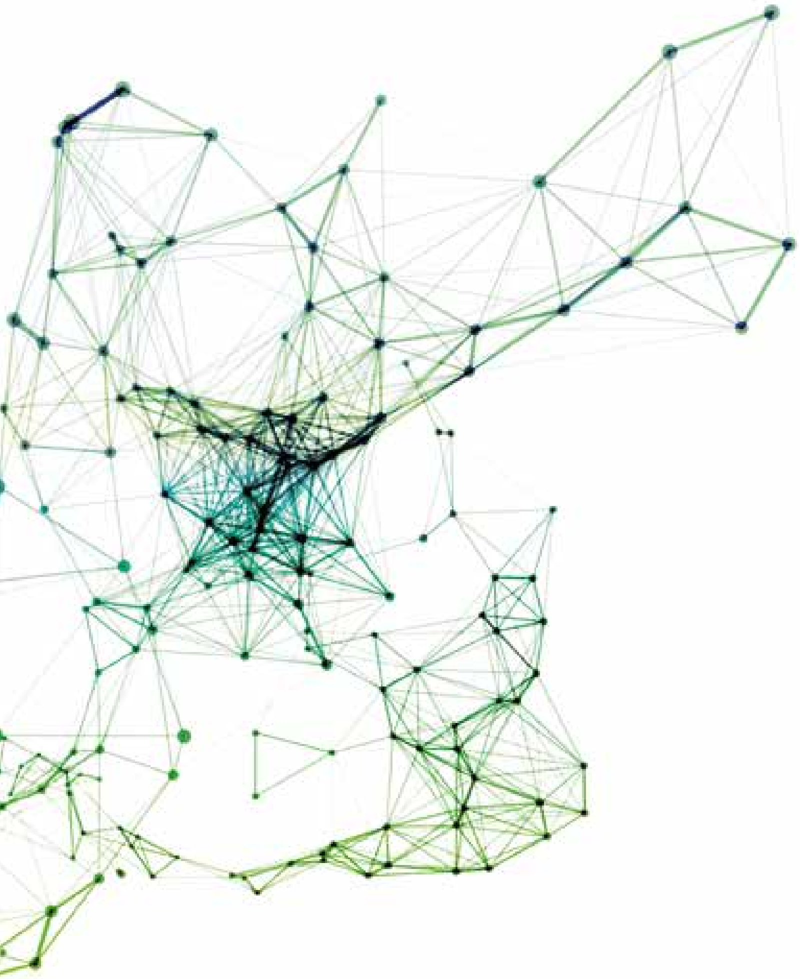
Another important issue in Deep Learning frameworks is hardware support. Beside the support for Intel CPUs and NVIDIA GPUs, the framework users have to completely rely on the efforts of a vendor to introduce support for their hardware in these frameworks. In PyTorch, over 60,000 lines of code are solely dedicated to NVIDIA GPUs, which gives an idea of the effort required to add support for a new device. Even when a vendor supports a framework, their code updates and changes usually do not get included in the official releases of the frameworks frequently, with hardware related code updates being subject to long delays. This translates into outdated framework variants, vendor specific branches with certain limitations, and difficult building procedures. Ultimately, this situation creates significant management complexity and limits the ability of users to take advantage of the most recent hardware and frameworks features.

In contrast, NEC DLP removes the need to add hardware support in the Deep Learning frameworks. This is due to a DLP's specialized abstraction layer, which transparently handles all requests between the framework and the devices. This allows NEC to rapidly develop support for new hardware, as it only needs to add support for the device in the DLP abstraction layer interface. For instance, NEC DLP can run all the TensorFlow, PyTorch or MXNet computations on the NEC SX-Aurora TSUBASA vector processor. This is configurable in any of the frameworks by just adding a single line of code, such as "dlp.device.set(dlp.device.VE, 0)", after importing the NEC DLP library.

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. To help the deployment model, some frameworks provide leaner libraries that can be included in applications. For instance, PyTorch provides LibTorch, which is a C++ interface capable of running previously trained PyTorch models. However, this package alone is over 700 MB, which would need to be distributed with the target application. Toolkits such as NVIDIA's TensorRT require the neural network to be explicitly converted into their own format, facing again significant engineering efforts and compatibility issues, such as missing layer implementations.

NEC DLP supports the deployment of the neural network models for any supported hardware. Using a deployment function “dlp_model.deploy(target=dlp.target.linux_shared, device=dlp.device.ve, name=’predict_model’)” is all that is needed to generate a shared Linux library that provides the function “predict_model(input, output)” that runs the network on the 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.



NEC LxFS-z for HPC Storage

Supreme Performance Combined with Highest Reliability

The Lustre file system is an open source, parallel file system that supports the requirements of leadership class HPC and enterprise environments worldwide. Lustre provides a POSIX compliant interface and scales to thousands of clients, petabytes of storage, and has demonstrated over a terabyte per second of sustained I/O bandwidth. Many of the largest and most powerful supercomputers on Earth today are powered by the Lustre file system, including over 60% of the TOP100 sites.

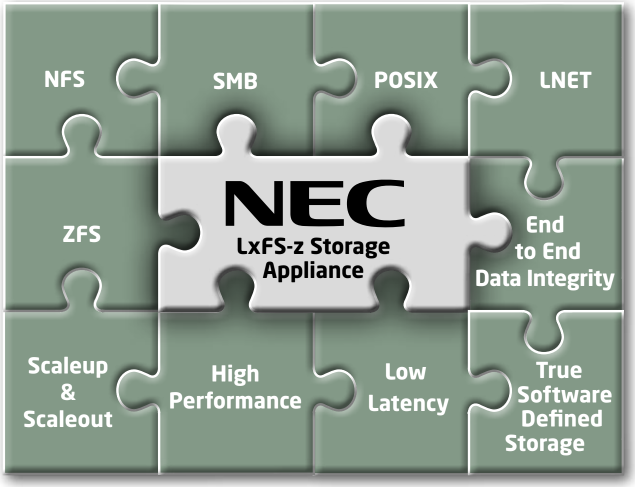
In conjuction with ZFS as an underlying filesystem, our NEC LxFS-z HPC storage appliance combine supreme performance with excellent reliability.

LxFS-z Overview

NEC LxFS-z Storage Appliance

In scientific computing the efficient delivery of data to and from the compute nodes is critical and often challenging to execute. Scientific computing nowadays generates and consumes data in High Performance Computing or Big Data systems at such speed that turns the storage components into a major bottleneck for scientific computing. Getting maximum performance for applications and data requires a high performance **scalable storage solution**. Designed specifically for High Performance Computing, the open source Lustre parallel file system is one of the most powerful and scalable data storage systems currently available. However, the managing and monitoring of a complex storage system based on various hardware and software components will add to the burden on storage administrators and researchers. **NEC LxFS-z Storage Appliance** based on open source Lustre customized by NEC can deliver on the performance and storage capacity needs without adding complexity to the management and monitoring of the system.

NEC LxFS-z Storage Appliance is a true **software defined storage** platform based on open source software. NEC LxFS-z Storage Appliance relies on two pillars: Lustre delivering data to the frontend compute nodes and ZFS being used as filesystem for the backend, all running on reliable NEC hardware. As scientific computing is moving from simulation-driven to **data centric computing** data integrity and protection is becoming a major requirement for storage systems. Highest possible data integrity can be achieved by combining the RAID and caching mechanisms of ZFS with the features of Lustre.

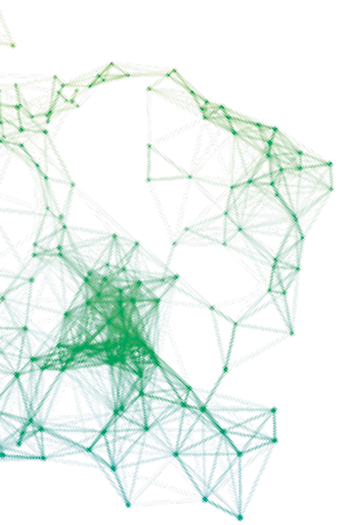


Highlights

- ➔ Lustre based parallel file system appliance for scientific computing
- ➔ Software defined storage solution based on ZFS for backend storage
- ➔ Complete storage solution, delivered with software stack fully installed and configured
- ➔ NEC SNA Storage Systems for best high-density, performance and reliability characteristics
- ➔ Fully redundant hardware and high-availability software suite for always-on operation
- ➔ NEC LxFS-z Storage Building Block concept for cost efficient system configuration matching performance and capacity demands
- ➔ Support for high-speed access via external protocols including SMB/CIFS and NFS
- ➔ Built-in granular monitoring of all system components to ease maintenance and maximise uptime
- ➔ Hadoop adapter for Lustre for seamless integration with Hadoop Infrastructures
- ➔ MapReduce adapter for HPC available to integrate in Big Data Analytics environments
- ➔ NEC support for both hardware and software

Lustre Architecture

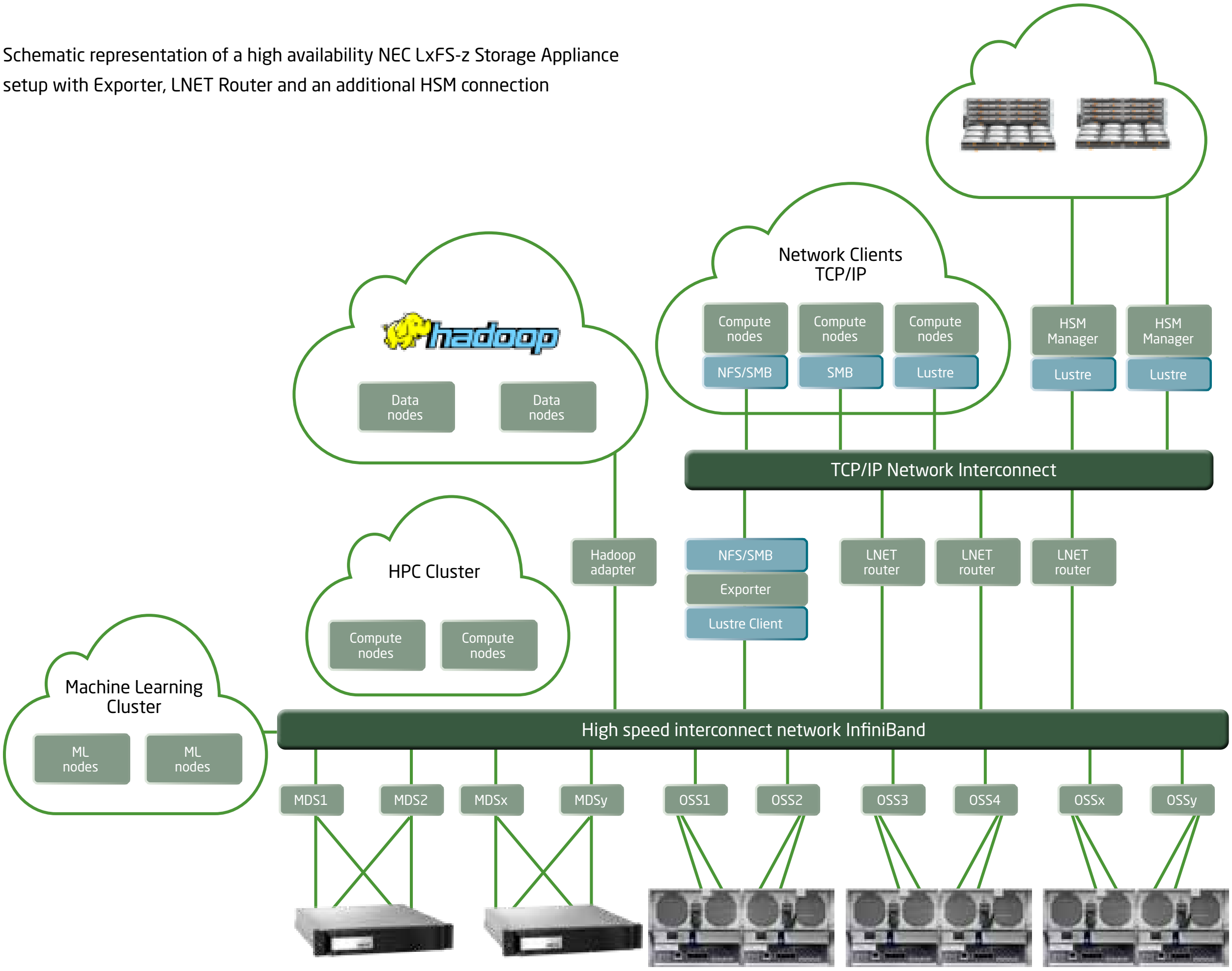
Lustre is an object based high performance parallel file system which separates file metadata management and actual file data handling and stores them on different targets. File metadata like name, permissions, layout, is stored on **Metadata Targets** (MDT) and processed by **Metadata Servers** (MDS) while file data is split into multiple objects and stored on **Object Storage Targets** (OST) which are mounted on **Object Store Servers** (OSS). The file system capacity is the sum of the OST capacities. On client side Lustre presents a file system using standard **POSIX** semantics that allows concurrent and coherent read and write access. When a client accesses a file, it completes a filename lookup on the MDS. The MDS returns the layout of the file to the client. After locking the file on the OST, the client will then run one or more read or write operations on the file by delegating tasks to the OSS. Direct access to the file is prohibited for the client. This approach enhances scalability, ensures **security and reliability**, while decreasing the risk of file system corruption from misbehaving or defective clients.

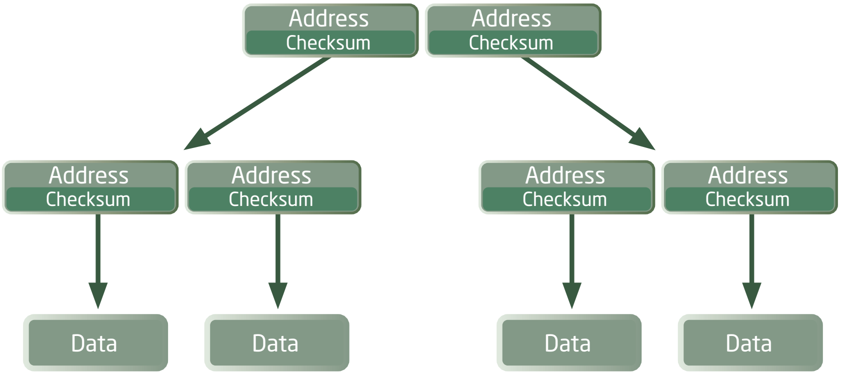


Servers and clients communicate through the Lustre Network **(LNET)** protocol, a network stack that is able to use TCP/IP and InfiniBand networks at performances close to their native speed. Lustre Clients can tolerate failures of servers as long as the targets can be mounted from other servers. **LNET Routers** can connect multiple InfiniBand or Ethernet networks and route LNET traffic between them. Lustre clients can re-export the file system through **NFS** or **SMB/CIFS** or act as data movers to/from a second tier HSM file system. The architecture enables scaling to thousands of OSTs, tens of thousands of client nodes, hundreds of petabytes of storage capacity and over a terabyte per second aggregated I/O bandwidth. Features like an **online file system check** allow to repair inconsistencies of the file system while the file system is up and running.

The NEC LxFS-z Storage Appliance scales not only in terms of bandwidth and performance but also in terms of connectivity. Using the Hadoop Adapter for Lustre a shared data repository for all compute resources can be established. The usage of data in place without import and export of data combined with a high performance low latency Lustre filesystem provided by the NEC LxFS-z Storage Appliance allows the creation of data centric workflows. You can have dedicated compute and data nodes or run a single compute cluster that can be used for variety of workloads. To support machine learning technologies, storage systems have to deliver throughput and performance at scale. I/O bottlenecks and massive data storage can be considered as a major challenge for machine learning and AI environments. The NEC LxFS-z Storage Appliance effectively reduces the problem of I/O bottlenecks and with its focus on high performance access to large data sets, NEC LxFS-z Storage Appliance combined with burst buffers and flash devices can be considered a valid solution to full scale machine learning systems.

Schematic representation of a high availability NEC LxFS-z Storage Appliance setup with Exporter, LNET Router and an additional HSM connection

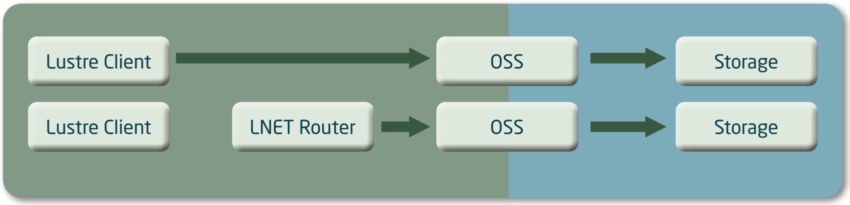




ZFS Solution for Data Integrity with Lustre

ZFS uses a **copy on write** transactional model for writing data. Blocks on disk with active data will never be overwritten and data will be always consistent on disk and a snapshot of the data can happen at no cost and without any performance impact anytime.

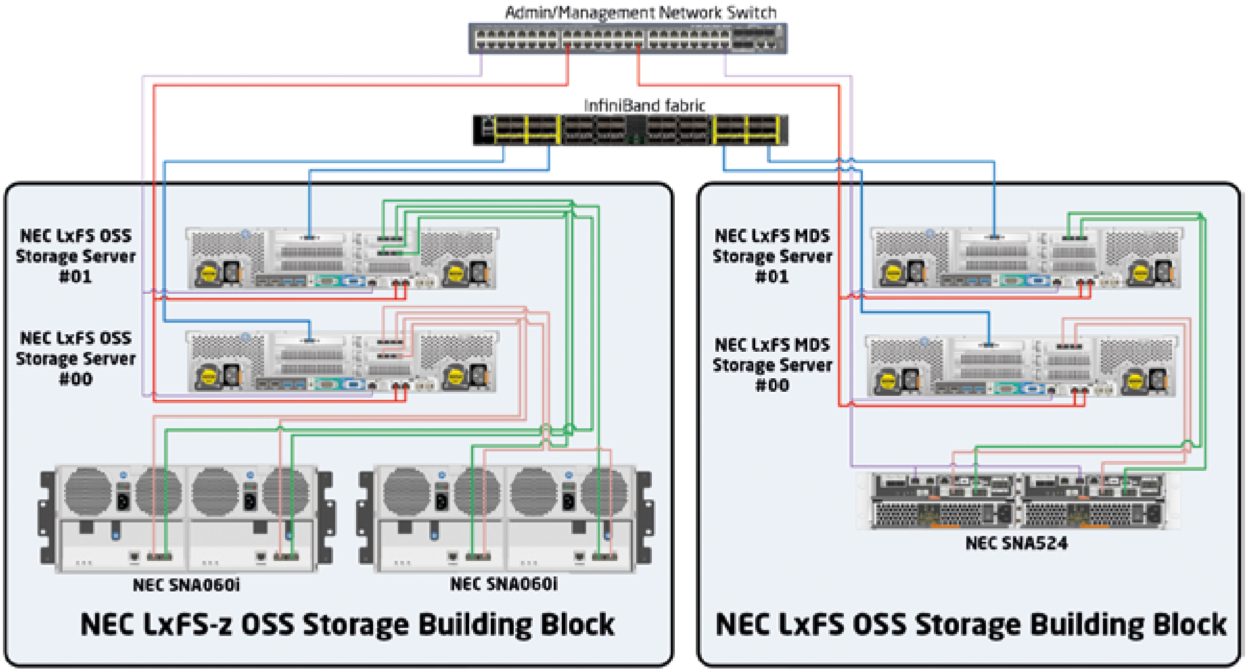
One of the key features of ZFS is the extremely powerful **software RAID engine** that allows single, dual, and even triple parity raid configurations. An important objective in the development of ZFS was to eliminate expensive hardware RAID controllers for building enterprise class storage solutions. The so-called RAID-Z software RAID implementation of ZFS has several outstanding features. To prevent the so-called "RAID-5 write hole" which can also happen when using RAID 6, RAID-Z uses variable-width RAID stripes resulting in all writes being full stripe writes. Full stripe writes guarantee not only data protection, they also greatly improve write performance. In combination with a highly tuneable and reliable I/O scheduler ZFS outperforms most of the hardware RAID-controller based storage systems. Intelligent caching algorithms greatly improve performance of a ZFS based system. Conceptually ZFS differentiates three caching methods. The **Adaptive Replacement Cache** (ARC) is the first destination of all data written to a ZFS pool, and as it is the DRAM of the server it is the fastest and lowest-latency source for data read from a ZFS pool. When the data is in the ARC. The contents of the ARC are balanced between the most recently used and most frequently used data. Level Two ARC (**L2ARC**) is an extension of the ARC based on SSD. The L2ARC Cache is a read cache to take the pressure from the ARC cache. The algorithms that manage ARC to L2ARC Migration work automatically and intelligently. The ZFS Intent Log (ZIL) is used to handle synchronous write/write operations that are required by protocol to be stored in a non-volatile location on the storage device before they can be T10-PI data protection, which protects only against silent data protection ensure data stability. ZFS can do this by using placing the ZIL on a mirror of enterprise grade write-optimized SSD. All writes (whether synchronous or asynchronous) are written into the DRAM based ARC, and synchronous writes are also written to the ZIL before being acknowledged. This is comparable to the concept of NVRAM used in a hardware RAID-controller. Under normal conditions, when ARC is flashed to drives, the data in the ZIL is no longer relevant. Especially the way ARC and hard disks work together is one of the keys to performance for ZFS backed systems.



NEC LxFS-z Storage Appliance Data integrity providing always consistent data from client to disk.

Common hardware raid based storage solutions offer only a small subset of possible methods to assure data integrity. Most common used is T10-PI data protection, which protects only against **silent data corruption** but for example can't protect against phantom writes or misdirected reads or writes. To counter data degradation ZFS uses **checksums** throughout the complete file system tree. Each block of data is checksummed and the checksum value is then saved in the pointer to that block rather than at the actual block itself. Next, the block pointer is checksummed, with the value being saved at its pointer, thus creating a Merkle tree resulting in a self-validating pool. Each time data are accessed the whole tree will be validated. A background or on demand process called **disk scrubbing** scans and verifies all data blocks against the checksums and automatically repairs damaged blocks.

The I/O path of the data is protected by two types of checksums from the client for data sent over the network to stable storage. Within Lustre a 32-bit checksum of the data read or written on both the client and OSS is computed, to ensure that the data has not been corrupted in transit over the network. In combination with the checksums generated by ZFS NEC LxFS Storage Appliance provides enterprise data integrity all along the data path from client to disk.


Admin/Management Network Switch
InfiniBand fabric
NEC LxFS OSS Storage Server #01
NEC LxFS OSS Storage Server #00
NEC SNA060i
NEC SNA060i
NEC LxFS-z OSS Storage Building Block
NEC LxFS MDS Storage Server #01
NEC LxFS MDS Storage Server #00
NEC SNA524
NEC LxFS OSS Storage Building Block

NEC LxFS-z Storage Building Blocks

Being a **software defined storage** appliance, the choice of components and the software configuration is crucial for usage in a production environment. Configuration of Lustre and ZFS can get complicated and error-prone, especially when hardware incompatibility causes issues. Therefore well-defined building blocks are the basic units of a NEC LxFS-z Storage Appliance. By design NEC LxFS-z Storage Appliance consists of two types of building blocks one for meta-data and one for object storage.

The metadata building block relies on two **NEC HPC128Rh-2** server systems and a dual controller NEC SNA524 raid system using high performance SAS drives in a RAID-10 setup.

The NEC SNA060i JBOD has redudant SAS extender to the server systems. From the disks to the servers, all connections are realized using state-of-the-art SAS-3 technology. Each NEC SNA060i Storage Enclosure is equipped with 60 high capacity NL-SAS disk drives. For each NEC SNA Storage System, four RAID-Z2 sets are configured with 14 NL-SAS disks each. The remaining four disk drives are configured as hot spare disks.

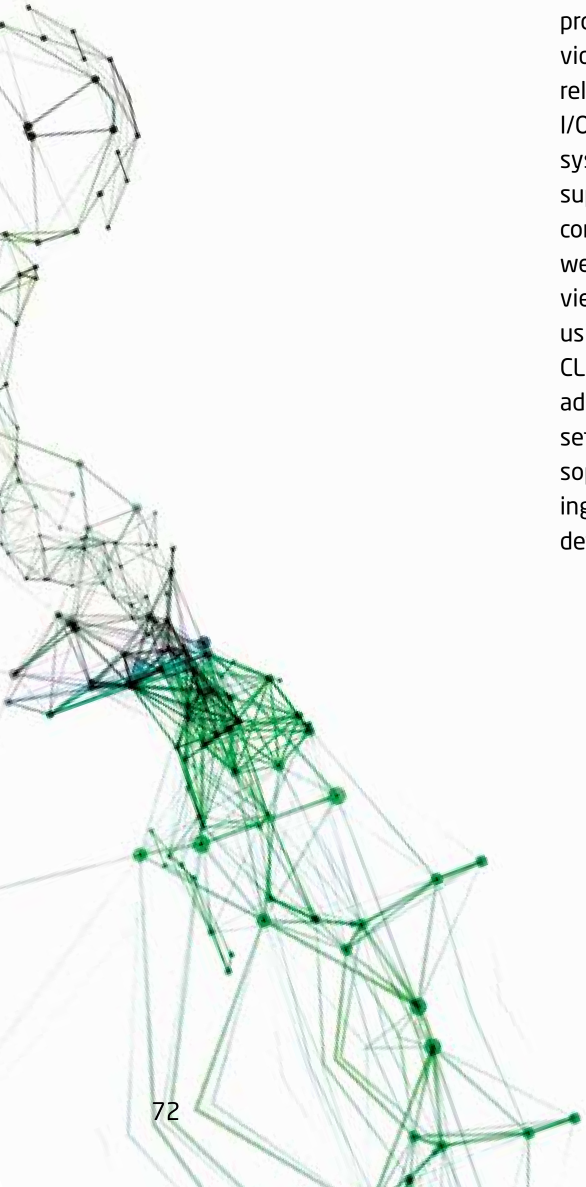
NEC LxFS-z Storage Appliance comes with a fully configured software stack **including high-availability**. The NEC LxFS-z Storage Appliance modular building block concept allows easy sizing and scaling of any I/O setup and data workflow. The building block concept allows to grow in capacity or bandwidth according to your demands. NEC LxFS-z Storage Building Blocks are delivering high performance and are due to the fully redundant configuration **without single points of failure** designed for always-on operation.

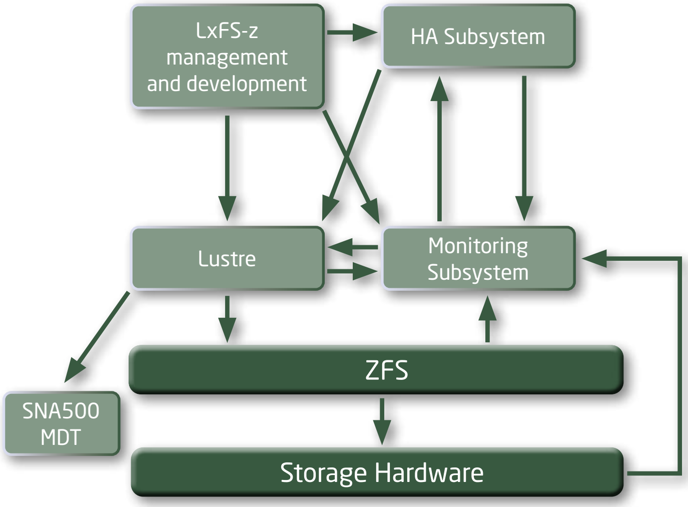
The NEC LxFS-z Storage Appliance comes with a completely configured software stack and an integrated monitoring solution. Based on the profound knowledge and a long-time operating experience with Lustre and ZFS NEC LxFS-z Storage Appliance is designed and configured for high bandwidth and reliable operation.

NEC LxFS-z Software Stack

Core of the software stack is the hardened version of the open source Lustre and ZFS on Linux. It is embedded into a framework of other software components providing reliability, management and monitoring features. Each building block runs an instance of the Pacemaker High Availability System which handles server failures gracefully without interruption of service.

In detail monitoring of important hardware and software components is provided by the health monitoring framework consisting of Nagios NEC-provided extensions. Performance monitoring data are collected by Ganglia and related NEC tools, that provide a large number of metrics about server loads, I/O, network and Lustre activity. Parameters and thresholds of the monitoring system are based on the long-time experience of NEC in building, running and supporting LxFS-z Storage Appliances. Monitoring data and system state can be conveniently visualised and accessed from anywhere by accessing the built-in webserver which provides highly configurable user customizable monitoring views. Based on alert rules alarm notifications will be sent in case of a failure using the preferred notification methods. The system comes with a set of CLI tools for easy management of the high-availability HA system and other administrative tasks, and for debugging and problem reporting. An automated setup deploys the NEC LxFS-z Storage Appliance in a reproducible way. This sophisticated software stack in combination with performance optimised building blocks makes the NEC LxFS-z Storage Appliance best of breed in software defined storage.





Block diagram of NEC LxFS-z Storage Appliance

NEC as a provider of Storage Appliances

The building blocks of the LxFS-z Storage Appliance are architected, integrated, tested, and optimized to work flawlessly together, thus cutting complexity and eliminating risks. This results in easier deployment and upgrades, and more efficient data and systems management. NEC not only provides hardware, but also optimal storage solutions based on know-how and experience of our employees. Consulting, benchmarking, implementation and support during all stages of a project from first design to 3rd level support are covered by NEC experts.

NEC has successfully implemented and is supporting LxFS-z Storage Appliances up to petabyte scale with a proven performance of more than 100 Gigabytes per second.

NEC BxFS-z for HPC Storage

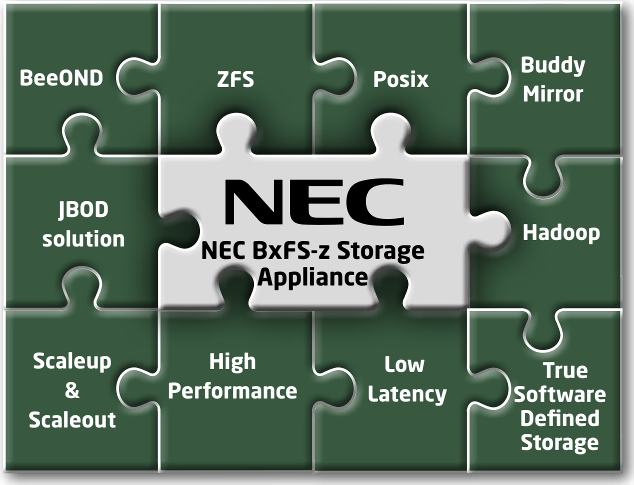
Highest Performance and Reliablility Made In Germany

BeeGFS (formerly FhGFS) is the leading parallel cluster file system, developed with a strong focus on performance and designed for very easy installation and management. If I/O intensive workloads are your problem, BeeGFS is the solution.

BxFS-z Overview

NEC BxFS-z Storage Appliance

With the ever increasing performance of modern processors and network technologies, the need for collected and processed data is also growing. In order to handle this huge amount of data and to ensure optimal performance in the calculation, the NEC BxFS-z Storage Appliance based on BeeGFS parallel file system has been developed. BeeGFS (also known as Fraunhofer Parallel Filesystem, formerly FhGFS) is a parallel file system, specially optimized for High Performance Computing (HPC). In addition to the very good scalability of the system, the NEC BxFS-z Storage Appliance attaches great importance to uncomplicated handling and high-availability combined with a maximum of flexibility and scalability. The NEC BxFS-z Storage Appliance is a true **software defined storage** platform based on open source software combining the scalability and features of BeeGFS with the data protection and RAID features of ZFS as underlying file system for the storage targets.



Highlights

- ➔ BeeGFS-based high performance parallel file system appliance
- ➔ Software Defined Storage solution relying on ZFS for backend storage
- ➔ On demand parallel file system BeeOND included
- ➔ Fully redundant high-availability setup
- ➔ NEC SNA high density JBOD solution
- ➔ Buddy Mirroring – built-in data replication feature for high-availability
- ➔ Complete storage solution, delivered with software stack fully installed and configured
- ➔ InfiniBand or Ethernet as high-speed interconnect with dynamic failover capability between different network topologies, with RDMA or RoCE support
- ➔ Multiple instances of BeeGFS can run in any combination on the same appliance
- ➔ Flexible data striping per directory or file
- ➔ NEC support for both soft- and hardware

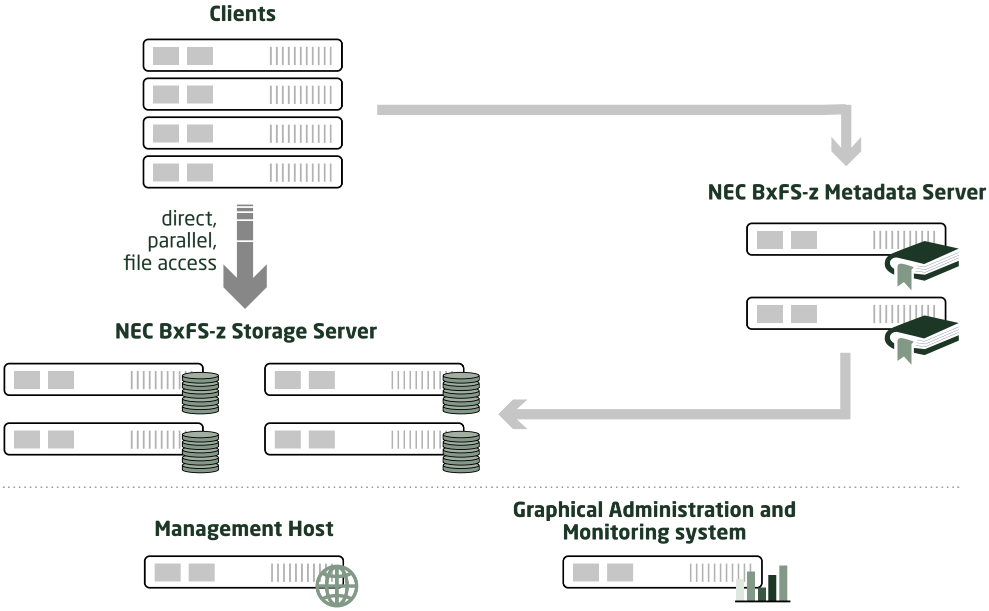
BeeGFS Architecture

From the beginning BeeGFS has been developed and optimized for **data throughput with a strong focus on scalability and flexibility.** Conceptually BeeGFS combines multiple storage servers to provide a highly scalable shared network file system with **striped file contents**. This way, it allows to overcome the tight performance limitations of single servers, single network interconnects, a limited number of hard drives etc. In such a system, high throughput demands of large numbers of clients can easily be satisfied, but even a single client or a single stream will benefit from the aggregated performance of all the storage servers in the system.

By design **BeeGFS separates metadata and file contents**. While storage servers are responsible for storing stripes of the actual contents of files, metadata servers do the coordination of file placement and striping among the storage servers and inform the clients about certain file details when necessary. When accessing file contents, BeeGFS clients directly contact the storage servers to perform file I/O and communicate with multiple servers simultaneously, resulting in **truly parallel access** to the file data. To keep the metadata access latency (e.g. directory lookups) at a minimum, BeeGFS can also distribute the metadata across multiple servers, so that each of the metadata servers stores a part of the global file system namespace. The following picture shows the system architecture and roles within an BeeGFS instance.

In the picture on the next page, all services are running on different hosts to show which services generally exist in a BeeGFS storage cluster. However, it is also possible to run any combination of BeeGFS services (client and server components) together on the same machines. Performance and capacity of a NEC BxFS-z environment can easily be scaled by adding NEC BxFS-z Storage Appliance building blocks to the level needed. Adding additional storage or metadata building blocks can be done without interrupting running data services.

Besides the three basic roles in BeeGFS (metadata service, storage service, client) there are two additional system services that are part of the NEC BxFS-z Storage Appliance. The first one is the management service, which serves as registry and watchdog for clients and servers, but is not directly involved in file operations and thus not critical for system performance. The second one is the optional administration and monitoring service (Admon), which provides a graphical frontend for installation and system status monitoring. Besides detailed storage performance metrics live per user and per client statistics are available using Admon. The NEC BxFS-z Storage Appliance will be delivered as a **turnkey solution** with all services preconfigured.


Clients
direct,
parallel,
file access
NEC BxFS-z Metadata Server
NEC BxFS-z Storage Server
Management Host
Graphical Administration and
Monitoring system

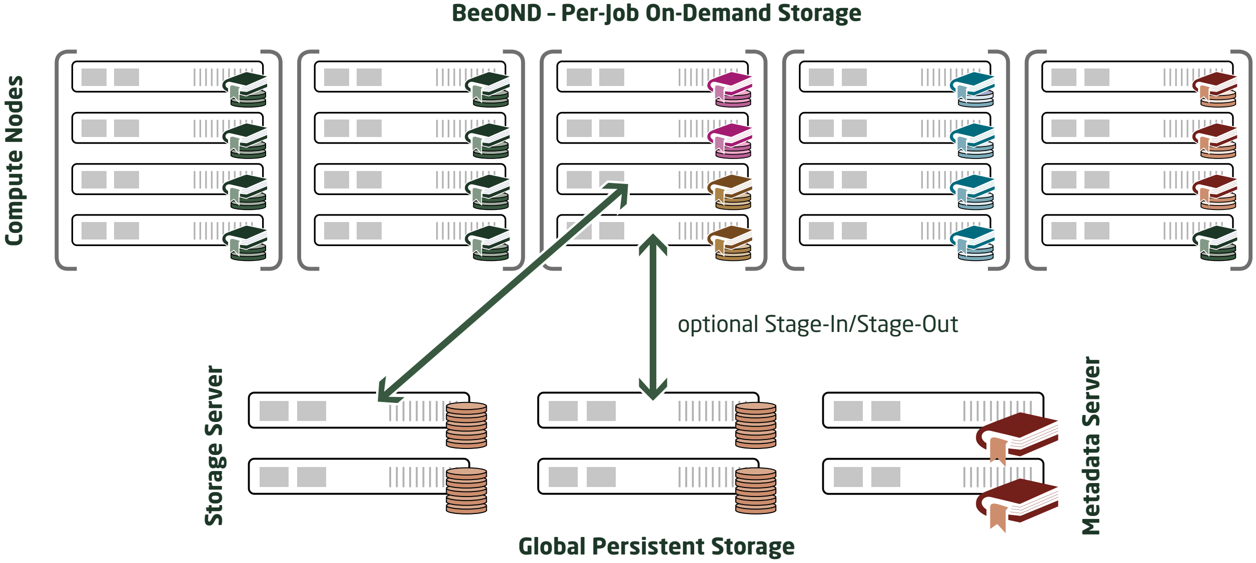
BeeGFS architecture

On-Demand Storage BeeOND

The NEC BxFS-z Storage Appliance provides a persistent global storage solution at the same time, it also offers the possibility of creating a temporary on demand parallel file system on the nodes in the compute cluster.

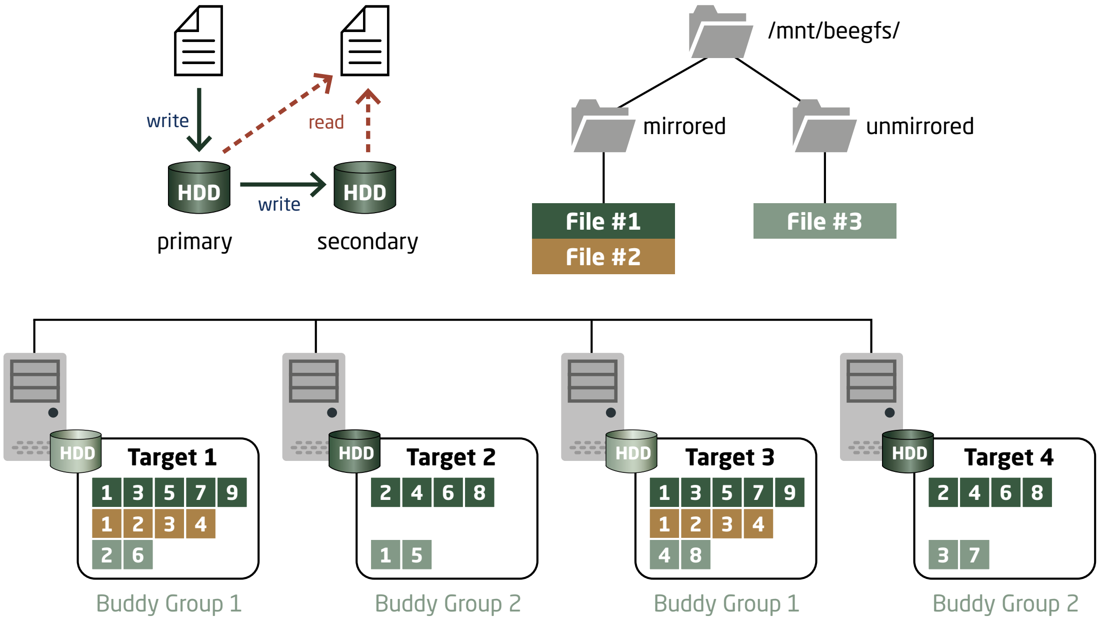
The efficient mapping from application data model to storage hardware is increasingly more complex. Therefore **BeeGFS on Demand (BeeOND)** was developed, to bring storage I/O closer to the computation layer. **BeeOND** allows on the fly creation of a complete, parallel file system instance on a given set of compute nodes with just one command. BeeOND is designed to integrate with cluster batch systems to create temporary, parallel file system instances on a per job basis on internal spinning or flash devices on the compute nodes, which are part of the compute job. This provides a very fast buffer while keeping much of the I/O load for temporary, random access files away from the global cluster storage. At the beginning of a job data can be staged from global persistent storage to the BeeOND parallel filesystem.

When the job is finished, the temporary parallel filesystem will automatically be shut down, data can be staged out using a parallel copy to persistent storage before the file system will be stopped. Typical BeeOND use cases would be for jobs that produce a lot of temporary data, read input x times or read and modify small chunks of data in-place. Not only for the use cases described BeeOND, offers a kind of **smart burst buffer** solution which can be easily implemented. In addition to this temporary parallel filesystem, the NEC BxFS-z Storage Appliance has built-in high-availability features.


BeeOND - Per-Job On-Demand Storage
Compute Nodes
optional Stage-In/Stage-Out
Storage Server
Metadata Server
Global Persistent Storage

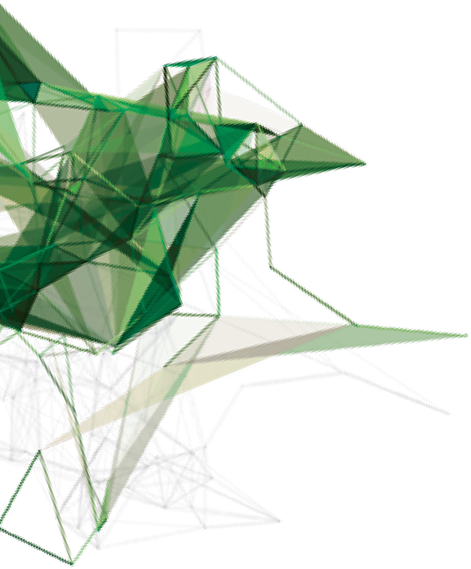
BeeOND principle of operation

High-availability for data and metadata plays a key role in scientific computing. Typically a shared storage architecture is used to keep availability of data and services high. NEC BxFS-z Storage Appliance follows this approach to handle storage server failures. If even the failure of a complete NEC BxFS-z Storage Appliance building block or a metadata server must be covered the built-in **BeeGFS Buddy Mirroring** can be used. With this feature enabled, data chunks are mirrored within primary and secondary targets the so-called Buddy Mirror Group. While reading is possible from both targets, modifying operations are sent to the primary target and forwarded to the secondary target. BeeGFS Buddy Mirroring automatically replicates data, handles storage server failures transparently for running applications, and provides automatic self-healing when a server comes back online, efficiently resyncing only the files that have changed while the machine was offline. BeeGFS Buddy Mirroring can be enabled on a per directory base, therefore targets with Buddy Mirroring enabled can also store non-mirrored chunks.



Combined with the sophisticated high-availability architecture of the NEC BxFS-z Storage Appliance, Buddy Mirroring for metadata in particular increases the availability of data without adding complexity. Buddy Mirroring offers a flexible solution to increase availability of data or metadata, as it can be enabled on a per-directory base.

The NEC BxFS-z Storage Appliance was designed with easy administration in mind. The graphical administration and monitoring system enables dealing with typical management tasks in a simple and intuitive way, while everything is of course also available from a command line interface. The monitoring system includes live load statistics even for individual users, storage service management and health monitoring.

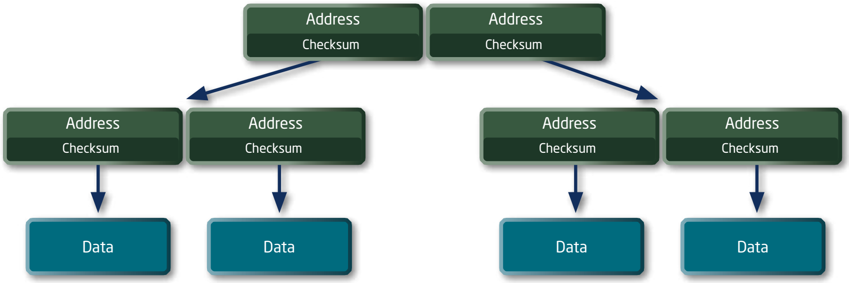


ZFS Solution for Data Integrity with BeeGFS

The NEC BxFS-z Storage Appliance uses ZFS as filesystem for the storage backend. ZFS applies a **copy on write** transactional model for writing data. Blocks on disk with active data will never be overwritten and data will be always consistent on disk and a snapshot of the data can happen at no cost and without any performance impact anytime. One of the key features of ZFS is the extremely powerful **software RAID engine** that allows single, dual, and even triple parity raid configurations. An important objective in the development of ZFS was to eliminate expensive hardware RAID controllers for building enterprise class storage solutions. The so-called RAID-Z software RAID implementation of ZFS has several outstanding features. To prevent the so-called "RAID 5 write hole" which can also happen when using RAID 6, RAID-Z uses variable-width RAID stripes resulting in all writes being full stripe writes. Full stripe writes guarantee not only data protection, they also greatly improve write performance. In combination with a highly tunable and reliable I/O scheduler, ZFS outperforms most of the hardware RAID-controller based storage systems. Intelligent caching algorithms greatly improve the performance of a ZFS-based system. Conceptually ZFS differentiates three caching methods.

The **Adaptive Replacement Cache** (ARC) is the first destination of all data written to a ZFS pool, and as it is the DRAM of the server, it is the fastest and lowest latency source for data read from a ZFS pool. When the data is in the ARC, the contents of the ARC are balanced between the most recently used and most frequently used data. Level Two ARC (**L2ARC**) is an extension of ARC based on

SSD. The L2ARC Cache is a read cache to take the pressure from the ARC cache. The algorithms that manage ARC to L2ARC Migration work automatically and intelligently. The ZFS Intent Log (ZIL) is used to handle synchronous write/write operations that are required by protocol to be stored in a non-volatile location on the storage device before they can be acknowledged to the host. ZFS can do this by placing the ZIL on a mirror of enterprise grade write-optimized SSD. All writes (whether synchronous or asynchronous) are written into the DRAM based ARC, and synchronous writes are also written to the ZIL before being acknowledged. This is comparable to the concept of NVRAM used in a hardware RAID-controller. Under normal conditions, when ARC is flashed to drives, the data in the ZIL is no longer relevant. Especially the way ARC and hard disks work together is one of the keys to performance for ZFS backed systems. Common hardware RAID-based storage solutions offer only a small subset of possible methods to assure data integrity. Most commonly used is T10-PI data protection, which protects only against **silent data corruption** but cannot protect against phantom writes or misdirected reads or writes. To counter data degradation ZFS uses **checksums** throughout the complete file system tree.

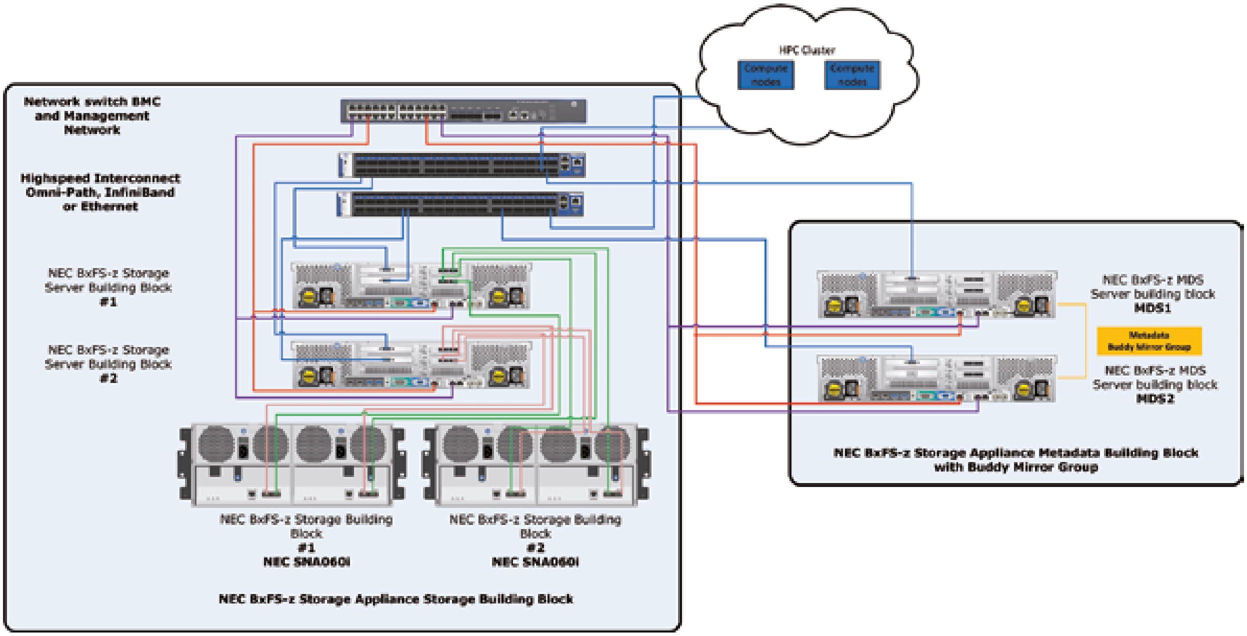


Each block of data is checksummed and the checksum value is then saved in the pointer to that block rather than in the actual block itself. Next, the block pointer is checksummed, with the value being saved at its pointer, thus creating a Merkle tree resulting in a self-validating pool. Each time data are accessed the whole tree will be validated, thus ensuring that only validated data will be read from stable storage. A background or on demand process called **disk scrubbing** is scans and verifies all data blocks against the checksums and automatically repairs damaged blocks. The underlying ZFS architecture ensures data integrity and protection at a high level making the NEC BxFS-z Storage Appliance best choice even for mission critical data. As NEC has a long-lasting experience with ZFS on Linux (ZOL) the selection and configuration of the ZOL version is optimally adapted to the hardware.

NEC BxFS-z Storage Appliance Building Blocks

With the NEC BxFS-z Storage Appliance being a **software defined storage** appliance, the choice of components and the software configuration is crucial for use in a production environment. Therefore well-defined building blocks are the basic units of the NEC BxFS-z Storage Appliance. The idea of building blocks is to combine proven components needed to build a fully functional BeeGFS environment. By design, the NEC BxFS-z Storage Appliance consists of two types of building blocks one for metadata storage and one for file data storage. The BxFS-z MDS Storage Server building block for metadata relies on **NEC HPC128Rh-1** server systems with internal high endurance SSD in a RAID 10 setup. To assure availability of metadata, two NEC BxFS-z MDS Storage Server can be configured as a buddy mirror group, thus replicating metadata. The NEC BxFS-z Storage Building Block has two components the storage server and the storage target. The performance-optimized NEC BxFS-z storage building block consists of two redundant **NEC HPC128Rh-2** server systems acting as storage server. Each storage server is connected redundantly to at least two high-density 60-bay **NEC SNA60i** JBODs for the storage target. The NEC SNA060i JBOD has redundant SAS extenders connecting the disks redundantly to the server systems. From the disk to the server, all connections are realized using state-of-the-art SAS-3 technology. Each JBOD is equipped with 60 high-capacity NL-SAS drives. For each JBOD, four RAID-Z2 sets are configured with 14 NL-SAS hard drives. The remaining four hard disks are configured as hot spare disks. The NEC BxFS-z Storage Appliance comes with a fully configured software stack **including high-availability**. The NEC BxFS-z Storage Appliance has built-in by design high-availability for BeeGFS. The default configuration of the NEC-BxFS-z Storage Appliance allows the failure of a storage or a metadata server without subsequent failure of the filesystem. The NEC BxFS-z Storage Appliance modular building block concept allows easy sizing and scaling of any I/O setup and data workflow. The building block concept allows to grow in capacity or bandwidth according to your demands.

NEC BxFS-z Storage Building Blocks deliver high performance and are due to the fully redundant configuration **without single points of failure** designed for always-on operation. Based on the profound knowledge and the long-time operating experience with ZFS and BeeGFS, the NEC BxFS-z Storage Appliance is designed and configured for high-bandwidth and reliable operation.


HPC Cluster
Compute nodes
Compute nodes
Network switch BMC and Management Network
Highspeed Interconnect Omni-Path, InfiniBand or Ethernet
NEC BxFS-z Storage Server Building Block #1
NEC BxFS-z Storage Server Building Block #2
NEC BxFS-z Storage Building Block #1 NEC SNA060i
NEC BxFS-z Storage Building Block #2 NEC SNA060i
NEC BxFS-z Storage Appliance Storage Building Block
NEC BxFS-z MDS Server building block MDS1
Metadata Buddy Mirror Group
NEC BxFS-z MDS Server building block MDS2
NEC BxFS-z Storage Appliance Metadata Building Block with Buddy Mirror Group

NEC as a provider of Storage Appliances

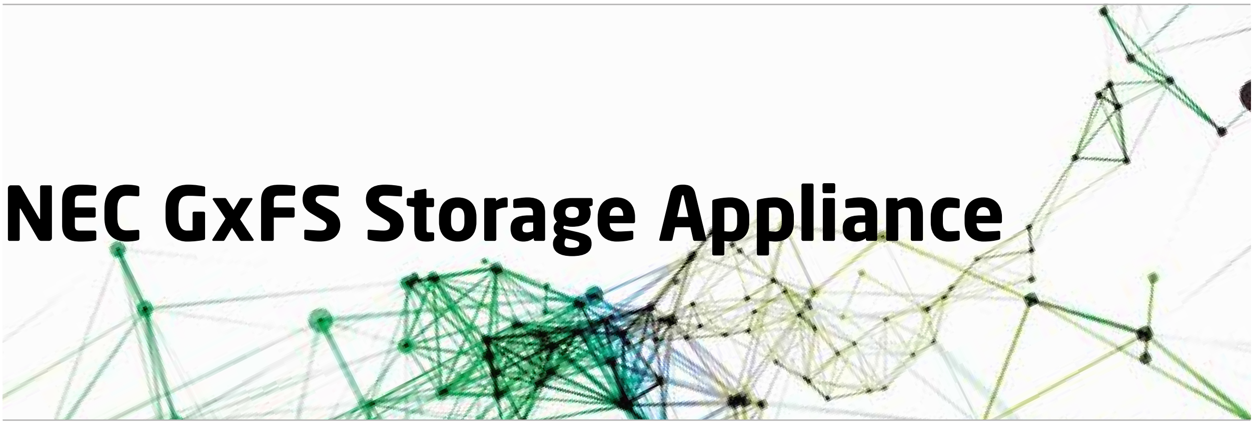
The building blocks of the NEC BxFS-z Storage Appliance are architected, integrated, tested, and optimized to work flawlessly together, thus cutting complexity and eliminating risks. This results in easier deployment and upgrades, and more efficient data and systems management. NEC not only provides hardware, but also optimal storage solutions based on know-how and experience of our staff. Consulting, benchmarking, implementation and support during all stages of a project from first design to 3rd level support are covered by NEC experts. NEC has successfully implemented and supports NEC BxFS-z Storage Appliances up to petabyte scale.

NEC GxFS Storage Appliance

Enterprise-Ready HPC Storage

The primary goal of HPC storage is performance: high bandwidth, high IOPS. But the more data are handled, the more important data reliability and enterprise-ready management capabilities become.

NEC GxFS HPC storage appliances provide a hitherto unparalleled combination of supreme performance and ease of use.



NEC GxFS Storage Appliance

Highlights

Following our HPC strategy NEC offers the different flavors of the GxFS Storage Appliance seamlessly integrated into the LXC3 framework. An integration of the NEC GxFS Storage Appliance using xCAT using the GxFS HA Admin Server is available as well.

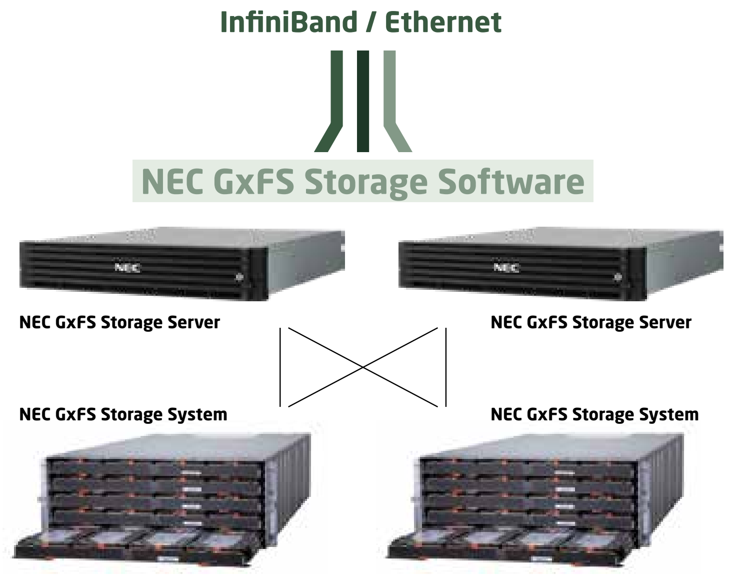
➔ NEC trusts on proven, reliable technology to ensure security for the most important ingredient to your IT infrastructure, the data itself! Due to that purpose the NEC GxFS Storage Appliance relies on the NEC Storage Systems from the SNA and the SSE product line with dedicated RAID controllers.

➔ As a new product line NEC is currently evaluating and certifying the Erasure Code based flavor of the NEC GxFS Storage Appliance.

➔ The NEC GxFS Storage Appliance follows the philosophy of providing Storage Building Blocks. These Building Blocks are providing the parallel file system functionality and are highly scalable based on the needs.

➔ Mature toolset to setup, operate and monitor the NEC GxFS Storage Appliance.

The NEC GxFS Storage Appliance follows a consequent building block approach. Each Building Block consists of two NEC GxFS Storage Server Systems configured as HA couple and a number of NEC SNA/SSE Storage Systems. The amount and layout of the NEC SNA/SSE Storage Systems will be tailor-made on the needs of the project. The Storage Systems are fully redundant connected to the Storage Server Systems using SAS-3 interface technology.



NEC GxFS Typical Storage Building Block

NEC GxFS Storage Appliance comes with built-in interfaces to attach the File Systems to industry standard Backup- and HSM systems. Furthermore, the NEC GxFS Storage Appliance is fully compatible with the variety of the Spectrum product line from IBM.

The NEC GxFS Storage Appliance scales up by adding building blocks to the configuration. The NEC GxFS Storage Appliance supports single name spaces up to a size of 8 exabytes and up to max number of 2^64 files per file system.

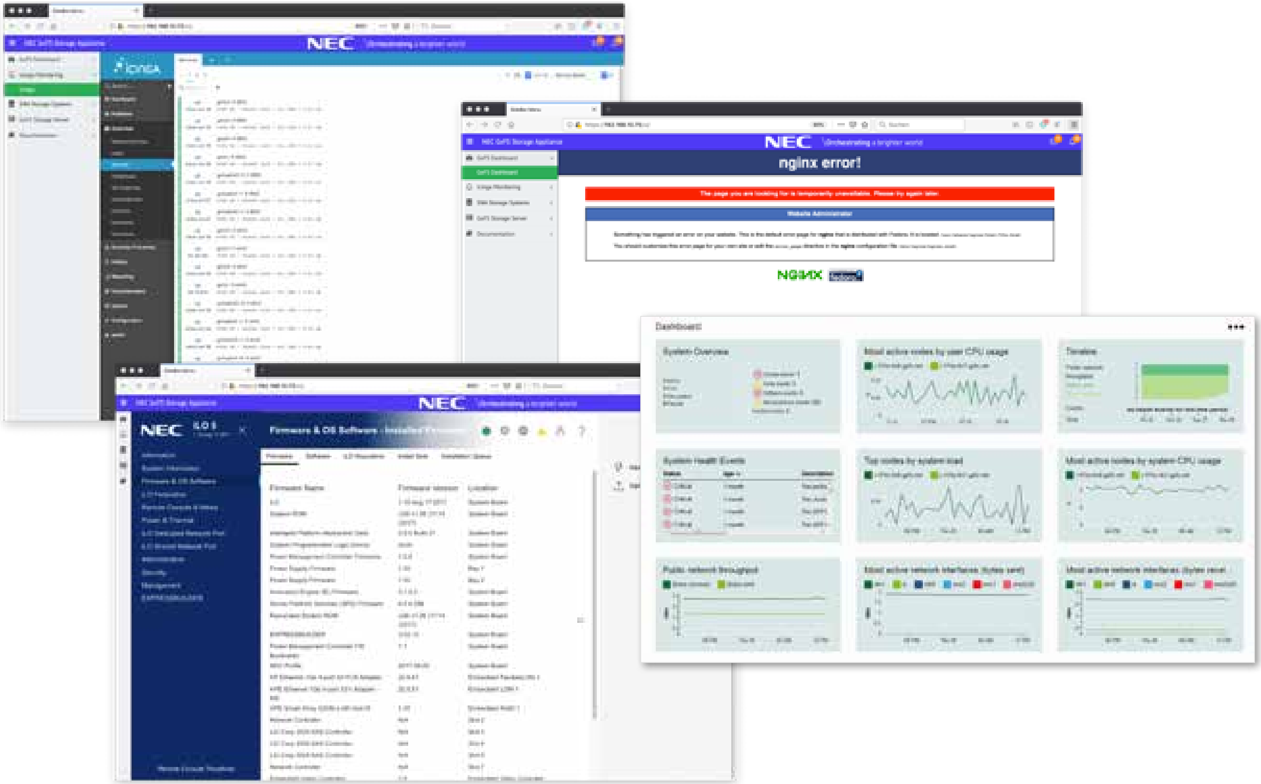
The interface technology used to connect the Storage Building Block with the outside world is using High Speed Interconnects like Mellanox InfiniBand (FDR/EDR/HDR) or Ethernet (RoCE). The NEC GxFS Storage Appliance is certified for the complete range of Mellanox HCAs. Mellanox HCAs can be used in both operation modes (InfiniBand and Ethernet).

NEC GxFS HA Admin Server

The NEC GxFS HA Admin Server concentrates the complete variety of monitoring or maintenance. The Admin Server consists of two servers configured as HA couple. The applications running on the HA admin server are running in Docker container, which ensures a seamless availability. In case one system fails for any reason the other one takes over all running applications with the same base of data. The data is distributed over both systems synchronously to ensure maximized reliability.

The following applications are running on the NEC GxFS HA Admin Server:

- Spectrum Scale GUI
- Spectrum Scale instance running the quorum node
- Icinga2 for monitoring
- Central access to Storage- and Server Systems via their Admin GUI
- xCAT provisioning tool


NEC
nginx error!
The page you are looking for is temporarily unavailable. Please try again later.
Website Administrator
NGINX
Dashboard
System Overview
Most active nodes by user CPU usage
Timeline
System Health Events
Top nodes by system load
Most active nodes by system CPU usage
Public network throughput
Most active network interfaces (bytes sent)
Most active network interfaces (bytes recv)
Firmware & OS Software - Installed Firmware
Firmware Name
Firmware Version
Location

IBM Spectrum Scale

Cognitive storage manages unstructured data for cloud, Big Data, Analytics, objects and more Enterprises and organizations are creating, analyzing and keeping more data than ever before. Those that can deliver insights faster while managing rapid infrastructure growth are the leaders in their industry. To deliver those insights, an organization's underlying storage must support both new era big data and traditional applications with security, reliability and high-performance. To handle massive unstructured data growth, the solution must scale seamlessly while matching data value to the capabilities and costs of different storage tiers and types. IBM Spectrum Scale™ meets these challenges and more. It is a high performance parallel file system for managing data at scale with the distinctive ability to perform archive and analytics in place. IBM Spectrum Scale enables the unification of virtualization, analytics, file and object use cases into a single scale-out storage solution. IBM Spectrum Scale can provide a single namespace for all of this data, offering a single point of management with an intuitive graphical user interface. Using storage policies transparent to end users, data can be compressed or tiered to the tape or cloud to help cut costs; data can also be tired to high performance media, including server cache, based on a heat map of data to lower latency and improve performance. Intelligent caching of data at remote sites ensures that data is available with local read/write performance across geographically distributed sites using Active File Management (AFM).

IBM Spectrum Scale is an enterprise-grade parallel file system that provides superior resiliency, scalability and control. Based on IBM General Parallel File System (GPFS™), IBM Spectrum Scale delivers scalable capacity and performance to handle demanding data analytics, content repositories and technical computing workloads. Storage administrators can combine flash, disk, cloud, and tape storage into a unified system that is higher performing and lower cost than traditional approaches. With thousands of customers and more than 15 years of demanding production deployments, IBM Spectrum Scale is a file system that can adapt to both application performance and capacity needs across the enterprise. By including IBM Spectrum Scale in the software-defined infrastructure, organizations can streamline data workflows, help improve service, reduce costs, manage risk and deliver business results today while positioning the enterprise for future growth.

Proven technology for high performance data management

IBM Spectrum Scale is full-featured, software-defined storage with management tools for advanced storage virtualization, integrated high availability, automated tiered storage and the performance to effectively manage very large quantities of file or object data. With the ability to independently scale in capacity, performance, protocols and resources, IBM Spectrum Scale is a clear leader in large, demanding environments. Organizations that may not have multiple petabytes of data today can start small with the confidence that IBM Spectrum Scale has already been tested in these environments.

Remove data related bottlenecks

Slow storage negatively impacts applications, delays schedules and wastes expensive infrastructure. IBM Spectrum Scale can speed time-to-results and maximize utilization by providing parallel access to data, shared disks and storage-rich servers, improving scalability for high performance workloads. IBM Spectrum scale is a parallel file system, where the intelligence is in the client and the client spreads the load across all storage nodes in a cluster even for individual files, while in traditional scale-out NAS one file can really only be accessed through one node at a time by an individual client. This parallel file system architecture allows IBM Spectrum Scale to seamlessly handle tens of thousands of clients, billions of files and yottabytes of data.

Simplify data management at scale

Part of the IBM Spectrum Storage™ family of solutions, IBM Spectrum Scale includes integrated management tools and an intuitive graphical user interface to help manage data at scale. IBM Spectrum Scale can span multiple storage environments and data centers to eliminate data silos and "filer sprawl." IBM Spectrum Scale can cognitively spread data across multiple storage devices optimizing available storage utilization, reducing administration and delivering high performance where needed. IBM Spectrum Scale has multiple deployment options and configurations to incorporate current NFS filers, block storage and storage-rich servers into a global namespace with universal access. The IBM Spectrum Scale file system supports interfaces for file (POSIX, NFS, CIFS), object (S3, SWIFT) or Hadoop Distributed File System (HDFS) for in-place analytics.

Empower global collaboration

IBM Spectrum Scale enables low latency read and write access to data from anywhere in the world using AFM distributed routing and advanced caching technology. AFM expands the IBM Spectrum Scale global namespace across geographical distances, providing fast read and write performance with automated namespace management. As data is written or modified at one location, all other locations get the same data with minimal delays. AFM leverages the inherent scalability of IBM Spectrum Scale, providing a high performance, location-independent solution that masks network failures and hides wide-area latencies and outages. These game-changing capabilities accelerate project schedules and improve productivity for globally distributed teams.

Cognitive data management

IBM Spectrum Scale can help improve performance, lower costs, add resiliency or simplify collaboration with algorithmic and policy-driven data movement, copying and caching. IBM Spectrum Scale catalogs data across multiple storage pools, including the cloud. It tracks usage profiles, storage latency and a broad range of standard and custom metadata from which data movement policies can be constructed. IBM Spectrum Scale is the caretaker of business-critical data with the ability to replicate, encrypt, compress, and distribute data across

different hardware platforms, systems and data centers. Armed with the knowledge of the data usage and the underlying storage, IBM Spectrum Scale curates data across multiple tiers of storage, including tape and cloud. The powerful data-aware intelligence engine can create optimized tiered storage pools by grouping devices flash, solid-state drive (SSD), disk or tape based on performance, locality or cost. Migration policies transparently move data from one storage pool to another without changing the file's location in the directory structure. Cognitive analysis of data usage patterns can help administrators pull data back up to higher performance tiers as needed. For example, administrators can create a rule that moves files out of the high-performance pool if the pool is more than 80 percent full reserving premium storage for use by active file data. The information lifecycle management toolset built into IBM Spectrum Scale helps simplify data management by enabling additional control over data placement. The toolset includes storage pooling and a high performance, scalable, rule-based policy engine.

End-to-end data availability, reliability and integrity

IBM Spectrum Scale provides system scalability, very high availability and reliability with no single point of failure in large-scale storage infrastructures. Administrators can configure the file system so that it automatically remains available if a disk or server fails. IBM Spectrum Scale is designed to transparently fail over metadata operations and other IBM Spectrum Scale services, which can be distributed throughout the entire cluster. For additional reliability, IBM Spectrum Scale supports snapshots, synchronous and asynchronous replication, and asynchronous error diagnosis while affected input/output (I/O) operations continue. IBM Spectrum Scale offers the protection of data at rest and secure deletion with file-level encryption.

IBM Spectrum Scale ECE (Erasure Code Edition)

IBM Spectrum Scale Erasure Code Edition (ECE) is a high performance scale-out storage for commodity servers. It's a new software edition of IBM Spectrum Scale family. ECE provides all the functionality, reliability, scalability, and performance of IBM Spectrum Scale on the customer's own choice of commodity servers with the added benefit of network-dispersed IBM Spectrum Scale RAID, and all of its features providing data protection, storage efficiency, and the ability to manage storage in hyperscale environments.

The Spectrum Scale RAID technology in ECE isn't totally new. It has been field-proven in over 1,000 deployed IBM Elastic Storage Server (ESS) systems. With the innovative network-dispersed IBM Spectrum Scale RAID adapted for scale out storage, ECE delivers the same capabilities on commodity compute, storage, and network components. Customers may choose their preferred servers that meet ECE hardware requirements with the best flexibility and cost.

ECE brings the value of enterprise storage based on commodity servers to the customers who are asking for it. It's composed of a set of homogeneous commodity servers with internal disk drives, typically NVMe and spinning disks. They are connected to each other with a high speed network infrastructure. ECE delivers all the capability of Spectrum Scale Data Management Edition, including enormous scalability, high-performance and enterprise manageability, etc. It also delivers the durable, robust, and storage-efficient with IBM Spectrum Scale RAID, e.g. distributes data across nodes and drives for higher durability without the cost of replication, end to end checksum identifies and corrects errors introduced by network or media, rapid recovery and rebuild after hardware failure, etc.

High-Performance Erasure Coding

ECE supports several erasure coding and brings much better storage efficiency, e.g. ~70% with 8+3p and ~80% with 8+2p Reed Solomon Code. Better storage efficiency means less hardware and cost, which can help customers to save a lot of budget without compromising system availability and data reliability. ECE erasure coding can better protect data comparing with traditional RAD-5/6, e.g. 3 nodes of fault tolerance with 8+3p and 11 or more nodes which can survive concurrent failure of multiple servers and storage devices. What's more, ECE implements high-performance erasure coding, which can be used in first tier storage. One of the typical use cases of ECE is to accelerate data processing typically with enterprise NVMe drives, which can deliver very high throughput. High-performance is a key differentiation comparing with other erasure coding implementations in distributed storage systems. Many of them can be used for cold data only.

NEC Storage Systems

NEC provides the powerful NEC SNA Storage Systems. The NEC SNA Storage line consists of two main lines.

- **NEC SNA Series**
- **NEC SSE Series**

Both Storage System Series are Hardware RAID based. There are differences in the density. All systems are very flexible in terms of host interface options and type of RAID presenting to the host.

NEC SNA Series

The NEC SNA Storage line consists of two main lines.

- **NEC SNA800 Series**
- **NEC SNA500 Series**

Both storage system lines are based on the same components, the difference is in the possible bandwidth of the controllers and in the number of supported drives.

NEC SNA800 Series

- Provides a sustained bandwidth of up to 13 GB/s
- Can handle up to 480 drives, which results in a maximum gross capacity of 5,760 TB per Storage Array in a mix- and match configuration (based on 12 TB NL-SAS drives).

NEC SNA500 Series

- Provides a sustained bandwidth of up to 7 GB/s
- Can handle up 180 drives, which results in a maximum gross capacity of 2160 TB per storage array in a mix- and match configuration (based on 12 TB NL-SAS drives).

Achieve field-proven and reliable performance efficiency for modern enterprise applications

The challenge

Your enterprise relies on core storage applications that are critical to business success. You need consistent application performance and continuous availability so you can achieve business goals. You must have a proven storage system that works with your application software to deliver value with reduced complexity. Because your operations depend on these applications, they must have greater than 99.999% availability. For this you need proven storage purpose built for HPC environments.

The solution

Your enterprise must have storage that can meet your performance and capacity demands without sacrificing simplicity and efficiency. That is why the NEC SNAx00 Series was designed with the SANtricity OS adaptive caching algorithms, which address a large range of application workloads. Those workloads range from database, high IOPS, or bandwidth-intensive streaming applications to a mixture of workloads in a high-performance storage consolidation point. With fully redundant I/O paths, advanced data protection features, and extensive diagnostic capabilities, you can achieve greater than 99.999% availability, data integrity, and security.

These range from small systems in which NEC SNAx00 Series is the only storage in a mixed workload environment to several of the world's largest storage systems in HPC parallel file systems, database, data warehouse, and everything in between.

Dynamic Disk Pools

Dynamic Disk Pools (DDP) greatly simplify traditional RAID management by distributing data parity information and spare capacity across a pool of drives. That enables easier capacity expansion and greater protection. A key concept of DDP is the dynamic rebalancing of data during changes in the number of drives, whether adding drives or in the case of drive failure. Unlike a traditional RAID volume group's rigid configuration with a specific number of drives, Dynamic Disk Pools can optimize from a minimum of 11 to the maximum supported by the NEC SNAx00 system. By dynamically changing the number of physical drives in the pool, DDP improves data protection through dynamically rebalancing across the remaining (or additional) drives more quickly than traditional RAID while maintaining greater performance. This reduces exposure windows from days to minutes. Because drives are not getting any smaller and data needs are increasing, protection against drive failures is more important than ever. Dynamic Disk Pools eliminate the complexities of RAID management with no idle spares to manage, no reconfiguring of RAID when expanding, and a significantly reduced performance impact following failure of a drive or drives when compared to traditional RAID.

Balanced Performance

The NEC SNAx00 Storage System continues with its longstanding heritage of balanced performance designed to support any workload. High-performance file systems and data-intensive bandwidth applications benefit from NEC SNAx00's ability to sustain high read and write throughput. High-performance metadata operations in parallel filesystems benefit from its high IOPS and low latency. Regardless of the application workload, NEC SNAx00 is designed to support maximum performance efficiency.

Modular Flexibility

The NEC SNAx00 offers multiple form factors and drive technology options to best meet requirements. The ultra-dense 60-drive system shelf supports up to 720 TB in just 4U. It is perfect for environments with vast amounts of data and limited floor space. Its 24-system shelf combines low power consumption and exceptional performance density with its cost-effective 2.5-inch drives. Both shelves support NEC SNAx00 controllers or can be used for expansion, enabling optimized configurations that best meet performance, capacity, or cost requirements.

Flexible Interface Options

The NEC SNAx00 supports a complete set of host or network interfaces designed for either direct server attach or network environments. With multiple ports per interface, the rich connectivity provides ample options and bandwidth for high throughput. The interfaces include quad-lane SAS, iSCSI, FC, and Infiniband to connect with and protect investments in storage networking.

Maximum Storage Density

Today's storage must keep up with continuous growth and meet the most demanding capacity requirements. The NEC SNAx00 is purpose-built for capacity intensive environments requiring optimal space utilization and reduced power/ cooling requirements. Its ultra-dense 60-drive 4U disk shelf provides industry-leading performance and space efficiency that reduce rack space by up to 60%. Its high-efficiency power supplies and intelligent design can lower power use up to 40% and cooling requirement by up to 39%.

High Reliability: No Scheduled Downtime

The NEC SNAx00 storage system delivers high-speed, continuous data access. With over 20 years of storage development behind it, the NEC SNAx00 is based on a field-proven architecture designed to provide high reliability and greater than 99.999% availability with appropriate configurations and service plans. Keeping data accessible through redundant components, automated path failover, and online administration, (including online SANtricity OS and drive firmware updates), simplifies management and maintains organizational productivity. Its advanced protection features and extensive diagnostic capabilities deliver high levels of data integrity, including Data Assurance (T10-PI) to protect against silent drive errors.

Intuitive Management

NetApp SANtricity Storage Manager software offers extensive configuration flexibility, which allows optimal performance tuning and complete control over data placement. With its dynamic capabilities, SANtricity software supports on the fly expansion, reconfigurations, and maintenance without interrupting storage system I/O.

Application Integration

NEC SNA Series products have been deployed and used in today's most popular application environments, such as VMware®. and Microsoft®. Exchange. It also is used with databases such as Oracle® Databases, Microsoft SQL Server®, and others. The system integrates into any environment with its configurable options. It also meets the demands of transactional applications, in which sustaining performance is critical. Design can lower power use up to 40% and cooling requirements by up to 39%.

NEC SNA800 and NEC SNA500 Series TECHNICAL SPECIFICATION

All data in this table applies to dual-controller configurations

	SNA860	SNA824	SNA560	SNA524
Form factor	4U/60 drives (both 2.5" and 3.5")	2U/24 drives (2.5")	4U/60 drives (both 2.5" and 3.5")	2U/24 drives (2.5")
Maximum raw capacity	720 TB 5.7 PB with expansion shelves (using 12 TB drives)	76.8 TB (using 3.2 TB SSD drives) 5.7 PB with expansion shelves (using 12TB drives)	720 TB system shelf 2.1 PB with disk shelves (using 12 TB drives)	76.8 TB system shelf 1.4 PB with disk shelves (using 3.2 TB and 10 TB drives)
Maximum drives	480 with 60-drive shelves 120 SSDs (20 SSDs per 60-drive shelf)	480 with mixed shelves 120 SSDs (20 SSDs per 60-drive shelf)	180 with 60-drive shelves 120 SSD limit	180 with mixed shelves 120 SSD limit
Drives supported	4/8/10/12 TB NL-SAS 10TB NL-SAS FIPS 900GB, 1.2/1.8TB SAS 1.8 TB SAS FIPS 800GB, 1.6/3.2 TB SSD 1.6 TB SSD FIPS	900GB, 1.2/1.8 TB SAS 1.8 TB SAS 10K FIPS 800 GB, 1.6/3.2TB SSD 1.6TB SSD FIPS	4/8/10/12 TB NL-SAS 10 TB NL-SAS FIPS 900GB, 1.2/1.8 TB SAS 1.8 TB SAS FIPS 800 GB, 1.6/3.2 TB SSD 1.6 TB SSD FIPS	900 GB, 1.2/1.8 TB SAS 1.8 TB SAS 10K FIPS 800 GB, 1.6/3.2TB SSD 1.6 TB SSD FIPS
Included host I/O ports	4 ports 16 Gb FC or 4 ports 10 Gb iSCSI (optical) or 4 ports 10 Gb iSCSI (copper)		4 ports 16 Gb FC or 4 ports 10 Gb iSCSI (optical) or 4 ports 10 Gb iSCSI (copper)	
Optional host I/O ports	8 ports 16 Gb FC 8 ports 10 Gb iSCSI (optical) 4 ports 10 Gb iSCSI (copper) 8 ports 1 2Gb SAS		8 ports 1 6Gb FC 8 ports 10 Gb iSCSI (optical) 4 ports 10 Gb iSCSI (copper) 8 ports 12 Gb SAS	
System maximums	Hosts/partitions: 512 Volumes: 2,048 Snapshot copies: 2,048 Mirrors: 128		Hosts: 256 Volumes: 512 Snapshot copies: 512 Mirrors: 32	
Dimensions	**SNA860 System Shelf** **SNA060c Disk Shelf**	**SNA824 System Shelf** **SNA024c Disk Shelf**	**SNA560 System Shelf** **SNA060c Disk Shelf**	**SNA524 System Shelf** **SNA024c Disk Shelf**
Height	6.87" (17.46cm)	3.34" (8.48cm)	6.87" (17.46cm)	3.34" (8.48cm)
Width	17.66" (44.86cm)	19" (48.26cm)	17.66" (44.86cm)	19" (48.26cm)
Depth	37.09" (94.23cm)	19" (48.26cm)	37.09" (94.23cm)	19" (48.26cm)
Weight	SNA860: 249.1lb (113kg) SNA060c: 247.4lb (112.2kg)	60.5lb (27.44kg)	SNA560: 249.1lb (113kg) SNA060c: 247.4lb (112.2kg)	60.5lb (27.44kg)

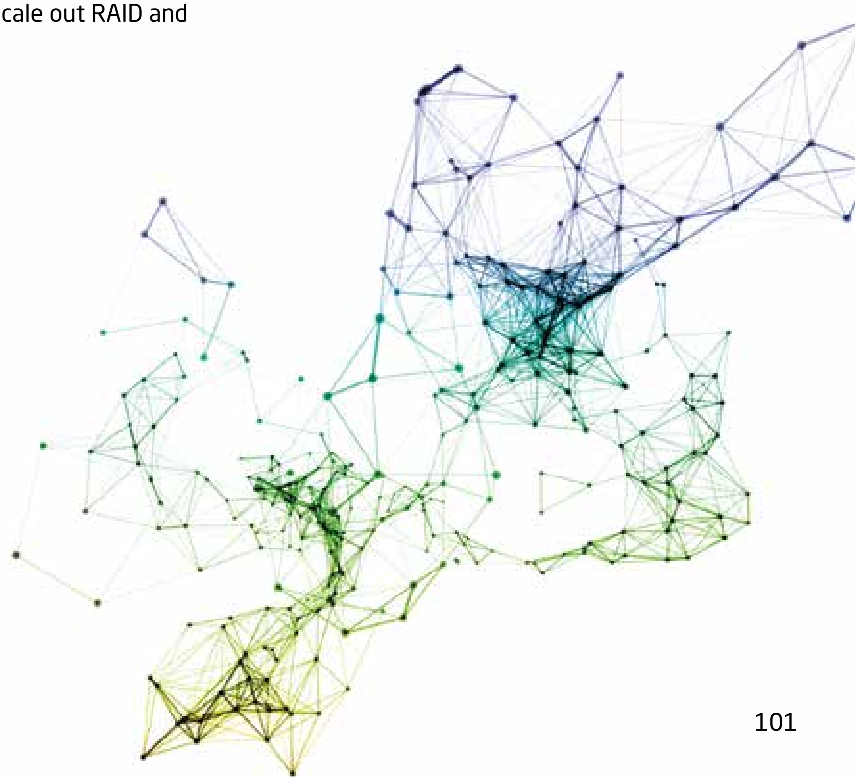
NEC SSE Series

Large data centers are increasingly limited by energy and space, and the NEC SSE storage systems make optimal use of these limited resources. The extremely dense, power efficient and performant enterprise class NEC SSE storage consists of two main lines.

- **NEC SSE400 Series**
- **NEC SSE500 Series**

Each model within the NEC SSE Storage Series use a common enclosure but is equipped with different controller typse to enable a broad range of use cases within the storage platform. These modular platforms with interchangeable modules not only reduce the number of spare parts but also takes burden from administrators handling large numbers of different storage systems.

- SSEx00-X models: Dual RAID-controllers with de-clustered RAID support
- SSEx00-A models: Dual x86 based Application Controllers for Software Defined Storage applications
- SSEx00-E models: Dual SAS Modules to scale out RAID and Application Controller based systems



NEC SSE400 Series

The ultra-dense NEC SSE400 Series is the industry's densest storage system, offering up to 1.7 PB of gross capacity in just 4U (based on 16 TB drives). The system greatly reduces the space and energy requirements and thus the infrastructure costs of the data center. Depending on the configuration, the bandwidth can reach up to 31 GB/s.

With ever increasing demand for higher capacity drives, the biggest challenge for high density enclosures is to provide stable drive performance across different conditions, specifically when enclosure fans runs at highest speed to accommodate for degraded conditions. It's a simple fact that cooling fans affect drive performance.

The acoustic energy coupled through the air (air coupling) does impact drives near each fan and impact the overall enclosure performance. The NEC SSE400 uses patented technology to block air coupling to maintain high performance even at maximum fan speed.

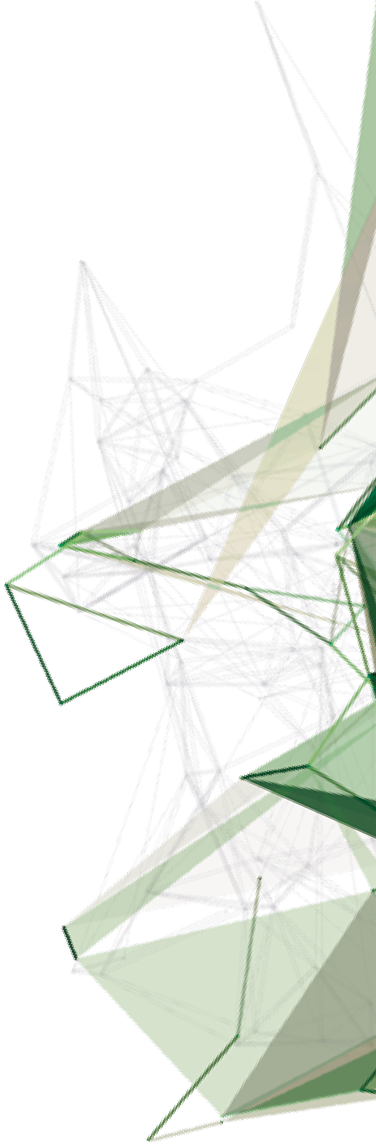
NEC SSE400E – 4U EBOD SAS enclosure with 106 x 3.5" drives

NEC SSE400-A Model

Modern software-defined storage solutions, such as the NEC QxFS and NEC GxFS ECE appliances, run on standard x86 servers, uses hard drives as well asflash devices, and provide data protection through data replication and erasure coding. The compact design of the NEC SSE400-A model combines a fully redundant 4U SAS enclosure and up to 100 drives with two dual-socket application servers and networking options such as InfiniBand or 100Gbps Ethernet. The combination of servers and storage in one chassis dramatically reduces hardware, cabling, and power consumption, resulting in a low total cost of ownership for the solution.

NEC SSE400-E Model

With the NEC SSE400-E model, the fully redundant 4U SAS enclosure can accommodate up to 106 drives and is used for capacity expansion of the NEC SSE400 A model, or external servers could be attached to the enclosure using the redundant SAS I/O modules.



NEC SSE500 Series

The dense NEC SSE500 Series provide up to 1.34 PB of gross capacity in 5U (based on 16TB drives) or 10.7 PB in 40U. The enclosure is fix-mounted in the rack and the two drawers allow easy access to the drives. Depending on the configuration, the bandwidth can reach up to 28 GB/s. The SSE500 series is optimized for installations where rack depth can’t be fully utilized for example in racks with back cooling doors which project inside the rack. Overall the SSE500 Series provides an excellent density and cost-for-performance.

NEC SSE524 – 2U SAS enclosure with 24 x 2.5" drives

NEC SSE584 – 5K SAS enclosure with 84 x 3.5" drives

NEC SSE500-A Model

Modern software-defined storage solutions, such as the NEC QxFS and NEC GxFS ECE appliances, run on standard x86 servers, uses hard drives as well as flash devices, and provide data protection through data replication and erasure coding. The compact design of the NEC SSE500-A model combines a fully redundant 5U SAS enclosure and up to 84 drives with two single-socket application servers and networking options such as InfiniBand or 100Gbps Ethernet. The combination of servers and storage in one chassis dramatically reduces hardware, cabling, and power consumption, resulting in a low total cost of ownership for the solution.

NEC SSE500-E Model

With the NEC SSE500-E model, the fully redundant 5U SAS enclosure can accommodate up to 84 drives and is used for capacity expansion of the NEC SSE500-A and SSE500-X models, or external servers could be attached to the redundant SAS I/O-modules.

NEC SSE500-X Model

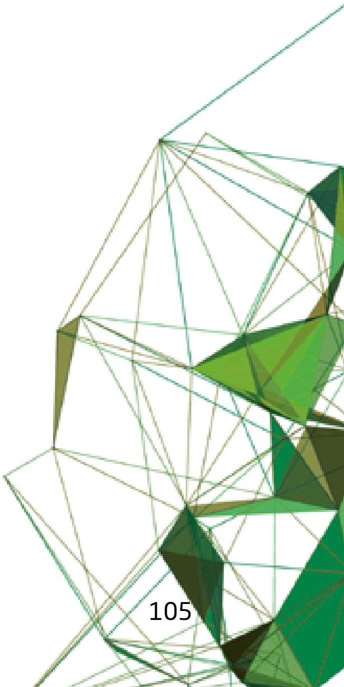
The NEC SSE500-X model combines a fully redundant 5U SAS enclosure and up to 84 drives with two hardware-based RAID controllers with eight SAS ports for host attachment. The SSE500-X Storage System can be expanded with SSE500-E models to a maximum of 336 drives with total of 5.3 PB gross capacity in just 16U. The RAID controllers include mirrored cache with battery backup and support RAID levels 0, 1, 3, 5, 6, 10, 50 as well as the de-clustered RAID functionality called ADAPT which is explained in a separate section below.

NEC SSE524-X and SSE524-X Models

The SSE524-X model use the same controllers and I/O-modules as the SSE500-X model but just in a 2U enclosure with 24x 2.5" drive slots and is expandable using SSE524-E models up to 240 drive slots. The system is typically used with flash drives for applications with high IOPS requirements for example Meta data servers. The RAID Controllers include mirrored caches with battery backup and support RAID levels 0,1,3,5,6,10,50 as well as the de-clustered RAID functionality called ADAPT which is explained in a separate section below.

ADAPT Data Protection Technology

RAID technology originally was optimized for use in environments with limited or finite storage capacity. But with traditional RAID types, the overall system performance can slow down dramatically during rebuilds when storage capacity increases. This is because data is striped across multiple drives along with parity information. If a drive in a RAID set fails, the controllers use the parity information to help reconstruct the data from a failed drive on a spare drive. Unfortunately, the speed at which the system can be rebuilt is tied to the performance of a single HDD. Today's high-capacity systems require new technology to replace data protection schemes that were developed based on much lower capacity systems.



ADAPT functions like RAID in that it is a data protection scheme, but its genius lies in dispersing the parity out to a number of drives. This structure enables the RAID controllers to take advantage of the combined performance of all those drives - versus being tied to a single drive. ADAPT simplifies data center management, allows for easy growth, ensures that end user applications have access to critical data, and dramatically reduces the time to successful fault tolerance (the ability to properly withstand an additional impact).

Thanks to ADAPT, the Autonomic Distributed Allocation Protection Technology developed by Seagate, the rebuild that took days now takes only minutes.

The table on the next page shows the calculated numbers for performance impacts and rebuild times for a RAID 6 (8+2) disk group versus an ADAPT disk group of 24, 56, and 106 drives. The key takeaway: the greater the ADAPT disk group size, the more the data protection benefits accelerate. This reduces the amount of time until the disk group can survive a third drive failure - from over 55 hours to just 25 minutes in an ADAPT group with 106 drives! (See bottom row in chart. Note that these are modeled numbers provided and results may vary based on actual workload. For more information consult the "ADAPT: Rapid-Rebuild Technology" Technology Paper from Seagate.)

Rebuild Time and Performance Under Failure

Metric	Traditional RAID 6 (8+P+Q)	24 Drive ADAPT	56 Drive ADAPT	106 Drive ADAPT
Perf impact, 1 drives down	-41%	-23%	-11%	-6%
Perf impact, 2 drive down	-62%	-37%	-20%	-12%
Rebuild 1 drive	55.5 hours	24 hours	10 hours	5.3 hours
Fault Tolerance: 3rd drive failure	55.5 hours	9 hours	1.5 hours	25 minutes

NEC SSE400 Series TECHNICAL SPECIFICATION

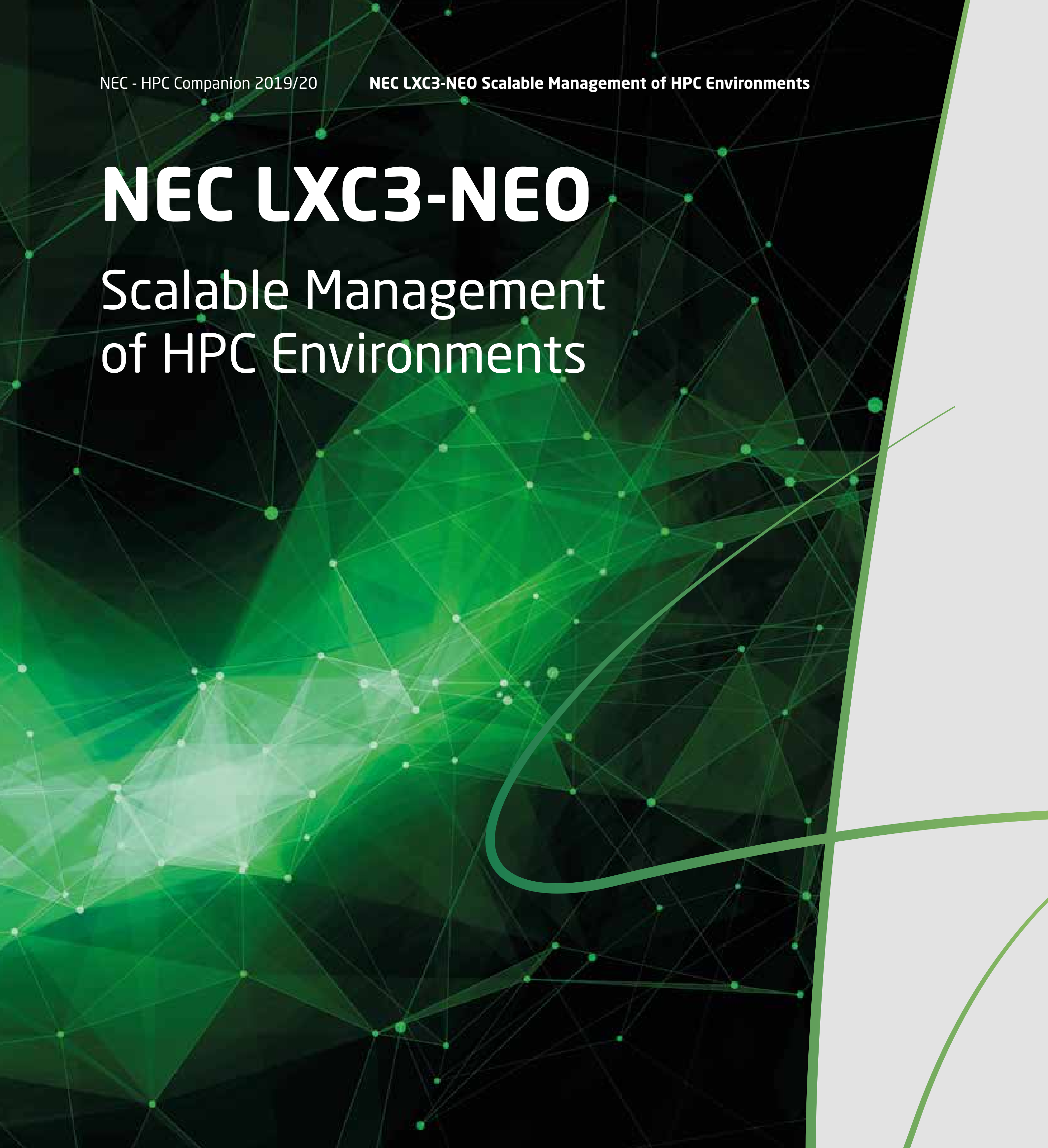
	SSE400X	SSE400E (EBOD)	SSE400A (AP)
Max drives/chassis	4 x 2.5″ + 96 x 3.5″	106 x 3.5″	4 x 2.5″ + 96 x 3.5″
Max drives/system		Up to 4 x SSE400E in daisy chain connected to the server	Up to 3 x SSE400E in daisy chain Up to 2 x SSE400E in star cable
I/O ports/ctrl	4 x 4 12 Gb/s SAS	4 x 4 12 Gb/s SAS	4 x 10 GbE SFP+ option EDR IB/100 GbE
Management ports/ctrl			1 x 1 GbE
CPU, Memory, PCIe	-	-	2 CPU sockets 12 DDR4 DIMM slots 2 low-profile x16 PCIe3
Height	4U/176 mm	4U/176 mm	4U/176 mm
Weight	150 kg	141 kg	150 kg
Depth (without cables)	1139 mm	1139 mm	1139 mm

NEC SSE500 Series TECHNICAL SPECIFICATION

	SSE500X (RAID)	SSE500E (EBOD)	SSE500A (AP)	SSE524X (RAID)	SSE524E (EBOD)
Max drives/chassis	84 x 3.5″	84 x 3.5″	84 x 3.5″	24 x 2.5″	24 x 2.5″
Max drives/system	336	Up to 4 x SSE500E (336 x drives) in daisy chain connected to the server	Up to additional 3 x SSE500E (336 x drives in total) in daisy chain connected to the SSE500A	Up to additional 10 x SSE524E (240 x drives in total) in daisy chain connected to the SSE524X	Up to additional 8 x SSE524E (192 x drives in total) in daisy chain connected to the server
I/O ports/controller	4 x 4 12 Gb/s SAS	3 x 4 12 Gb/s SAS	EDR IB/100 GbE	4 x 4 12 Gb/s SAS	3 x 4 12 Gb/s SAS
Management ports/ctrl	1	-	2	1	-
CPU, Memory, PCIe	-	-	1 CPU socket 4 x DDR4 DIMM slots	-	-
Height	5U/222 mm	5U/222 mm	5U/222 mm	2U/88 mm	2U/88 mm
Weight	135 kg	130 kg	135 kg	30 kg	24 kg
Depth (without cables)	981 mm	933 mm	981 mm	630 mm	630 mm

NEC LXC3-NEO

Scalable Management of HPC Environments



High performance computing clusters gain in performance not only since the built-in CPUs, memory and other hardware components are more powerful than ever, but also since the sheer size of HPC clusters grows. HPC clusters located in the top segment nowadays comprise several thousand compute nodes and the number is still increasing. This growth requires a rethinking about HPC cluster architectures that used to be common in the past. New ways to assemble, manage and administer such big machines are needed.

NEC LXC3-NEO Overview

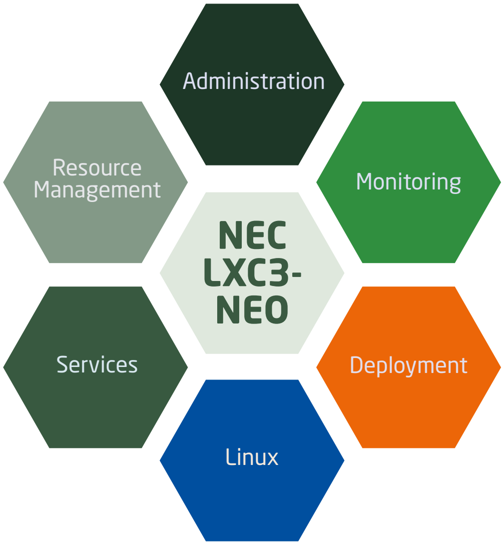
NEC LXC3-NEO Cluster Command and Control

LXC3-neo is NEC's new cluster command and control stack for the LX series high performance Linux clusters. It integrates nearly two decades of our cluster administration experience at HPC data centers of all sizes, and know-how from using and actively developing open source software with new ideas from research and development activities. LXC3-neo is the successor of NEC's LXC^3 software stack and has been newly developed from scratch, based on the latest technology borrowed from and battle-proven in the biggest data centers like Amazon AWS and Google. LXC3-neo is based on components that are thoroughly adapted to fit into the HPC world.

LXC3-neo contains all components that are necessary to make the administration and operation of a huge and complex HPC cluster as easy as possible:

- **Deployment/provisioning**: it supports stateless (operating system runs from RAM memory), stateful (operating system is installed on local hard drive) and compute nodes deployed in a hybrid way.
- **Cluster management tools:** various tools that ease the administration and usage of the cluster system.
- **Resource management:** job scheduler and batch system.
- **Monitoring** (performance and health) including **alerting** can be easily integrated on customer request.

The provisioning system of the cluster is originally based on Perceus 1.6, a well-established (but discontinued) system originating in the Warewulf project from Lawrence Berkeley National Labs. Perceus has been adapted and customized for LXC^3 by NEC and is now tightly integrated with other cluster components. LXC3-neo has been built from scratch but maintains the same battle proven provisioning scheme. The design goal was to gain best resource usage through high scalability (load balancing) and state of the art system up-time (high availability).



Provisioning System

The LXC3-neo provisioning system is image-based and supports stateless and stateful deployments of cluster nodes, allowing for completely diskless compute nodes, hybrid nodes with parts of the operating system (OS) on disk(s), parts in main memory or on an NFS network share, or full on-disk installations.

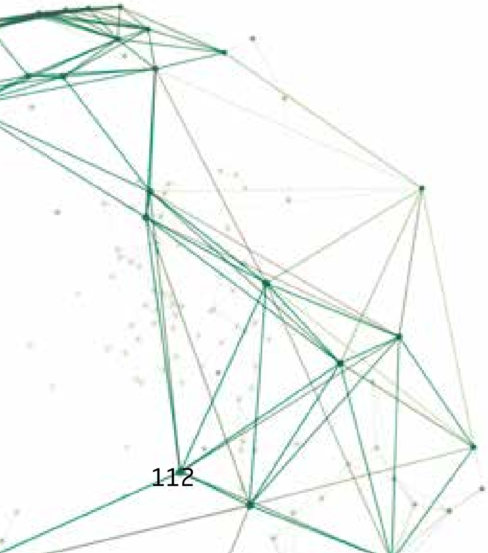
The image-based provisioning system is imposing but not enforcing an administration methodology for clusters. The main cluster node synchronization point is its central Virtual Node File System (VNFS), which allows to maintain a single system image for many cluster nodes. Node specific settings like network addresses and hostnames are auto-configured from the cluster description (Data Model), which is stored in a redundant database during the deployment of the compute node. The central administration paradigm is complemented by a procedural administration method through which compute nodes regularly check and pull configuration modules that can adjust their setup. Tracking changes within a VNFS image or within configuration modules can be easily done with the integrated versioning features.

LXC3-neo provides scripts to create VNFS images for common enterprise-grade Linux operating systems like RHEL, CentOS and SLES. Other Linux-based distributions can be integrated using a regularly installed system that will be used as a “golden client” source for the VNFS image.

Features

The following contains a list of features found in LXC3-neo.

- Single master or multiple master (3, 5, 7, … master nodes) scenarios possible. High availability and scalability will provide minimal downtime, maximize cluster availability and optimize resource usage.
- Automatic failover mechanism to automatically move system services and resources from a failed master to other machines.
- Completely symmetric master setup: Each master node contributes equally to the required resources like a distributed database (e.g. to store the Data Model) or shared disk storage (e.g. to hold the user’s home filesystem). No need for expensive hardware RAID system.
- Use different enterprise grade OS versions and distributions on master and compute nodes like Red Hat Enterprise Linux, CentOS or SuSE SLES.
- Repeatable setup procedure through fully automated installation procedure from bare metal to compute-ready. This is based on a cluster description file (Data Model) that contains every detail of the cluster and which is after the installation maintained in a distributed database.
- Ability to deploy stateless and stateful Operating System (OS) images (named VNFSes) to client nodes through a uniform two stage boot concept.
- Stateless deployments can be hybridized to minimize OS memory footprint by mounting parts of the operating system via NFS.



- ✔ Compute node grouping: Assign VNFSes to individual nodes or group of nodes. Usually compute nodes are grouped by common hardware features like containing GPU or Vector Engine (Aurora) cards.
- ✔ Create VNFS images using a simple two step procedure which pulls software packages from a local or remote repository.
- ✔ VNFSes can also be created using a regularly installed system as “Golden Image”.
- ✔ An automatic configuration step copies queueing system, provisioning system and other required configuration like NTP into a newly created VNFS to make it instantly ready for deployment.
- ✔ Principally any Linux distribution is as compute node operating system usable. Well tested and most requested is CentOS and Red Hat Enterprise Linux, but SuSE SLES and Scientific Linux are also possible.
- ✔ Export VNFSes from master nodes for backup purposes or import on other LXC3-neo installations.
- ✔ Clone VNFSes for test purposes and throw them away afterwards. The amount of VNFSes kept is only a matter of disk (shared storage) size.
- ✔ Simple standard procedure to administer, update and change VNFSes.
- ✔ Vector Engine and GPU card support easily addable through simple installation of tools and libraries in compute node image (VNFS).
- ✔ Compute node kernel updates can be done on the fly while compute nodes are running jobs through updating the VNFS image. The new kernel will get active after job termination and rebooting the compute node.
- ✔ Mechanism (script) to push small changes made in a particular VNFS to running compute nodes without rebooting the node.
- ✔ Single command line tool (with bash auto completion) to administer compute nodes, VNFSes and other configuration settings.
- ✔ VNFSes can be put under version control to track changes and roll misconfigurations back.
- ✔ Web-based GUI for simple VNFS and compute node management (not exhaustive).

- ✔ Automatic configuration (IP address, network mask, username and password) of BMC controller at boot time for all compute nodes, regardless if stateful or stateless.
- ✔ Preconfigured synchronization of system date and time on master nodes with external or customer provided NTP server.
- ✔ Automatic synchronization of compute node date and time with master node.
- ✔ Simple integration of customer provided user management system like NIS or LDAP.
- ✔ Automatic key management to allow passwordless login to compute and master nodes (on master nodes administrators only).
- ✔ Automatic export of all users $HOME directory on master node and mount on compute nodes.
- ✔ User friendly and open for extension and customization through LXC3-neo module concept (simple shell scripts) and Ansible (pull) integration.
- ✔ Use local compute node disks as scratch or swap (also while OS is stateless) space. Scratch can be erased or kept during reboots. Local disks will be partitioned and formatted (on a per VNFS basis) based on an XML description file.
- ✔ The same XML schema based configuration like for scratch space will be applied for stateful deployments.
- ✔ Boot compute nodes with UEFI firmware (no secure boot and no stateful GRUB boot yet) instead of legacy BIOS boot.
- ✔ Load the latest Intel Microcode updates to increase stability and security and to enable extended features even in completely stateless designs.
- ✔ Automatic procedure to use motherboard manufacturer’s provided EFI or Linux tools to update compute node firmware.
- ✔ LXC3-neo modules that set certain (per node, per VNFS or per group of nodes) optimizations before booting the final (READY state) kernel. This includes NUMA settings, ulimits and can be easily extended for optimal compute node performance.
- ✔ Support of extended file attributes (xattr) on stateless and stateful compute nodes. This includes ACLs, capabilities and security features which are implemented using xattrs on Linux.

- ✔ LXC3-neo provisioning can either use NFS or HTTP. In case hybridization is not used and user home directories are not mounted via NFS an NFS server is not required.
- ✔ LXC3-neo can be complemented with Ansible a well known and widely used configuration management tool. For best scalability LXC3-neo implements the ansible-pull scenario.
- ✔ Improved performance for command line usage (especially VNFS mount/umount) and compute node boot time by using all available cores for computational expensive operations like packing and unpacking archives.

Cluster Management

LXC3-neo comprises a set of tools to manage a cluster in a scalable and effective manner. These tools can be used to boot and power down the whole cluster, switch power of individual nodes and/or groups of nodes. It is also possible to soft-restart nodes, which is useful in case a node does no longer allow to remote login.

There are also tools that can be used to perform regular administration tasks on multiple nodes in parallel to ease the daily work of administrators. These tools are executed on the master node and perform the requested actions in parallel on the target nodes, collect the output of the commands and display it in an adequate way on the master node.

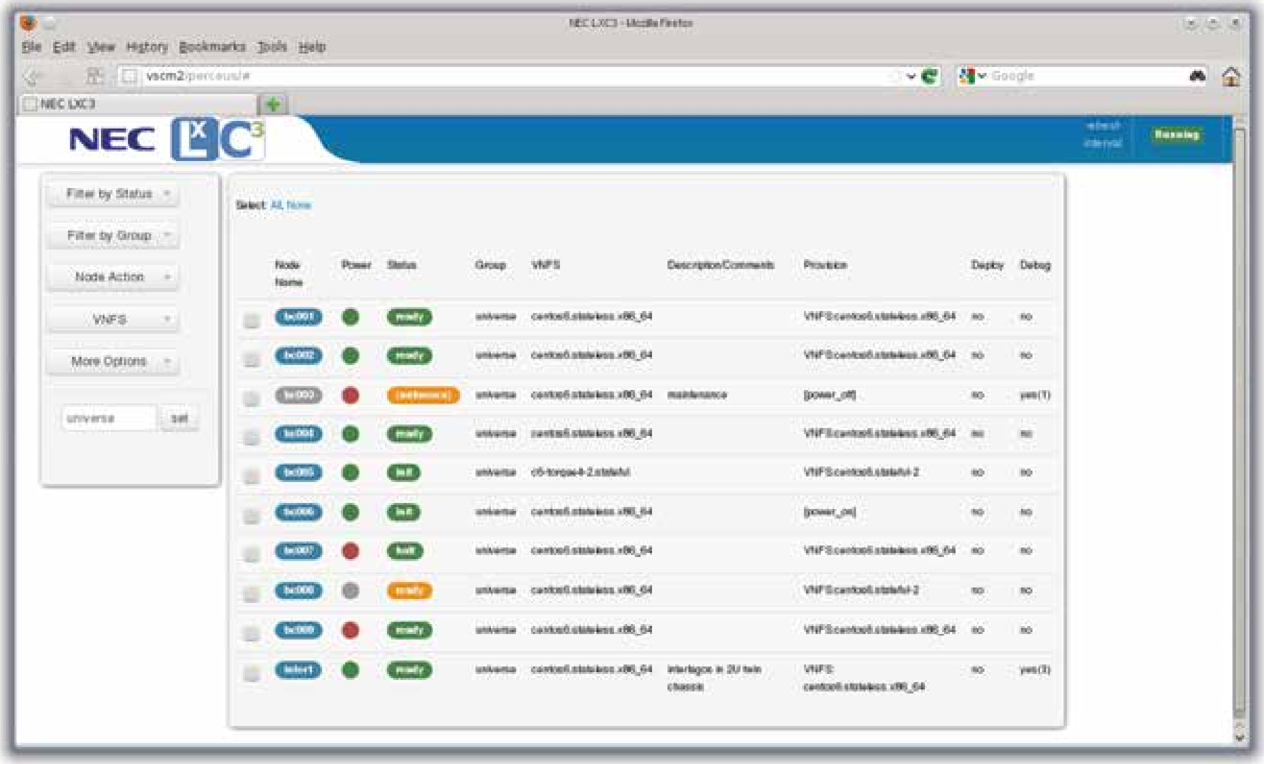
LXC3-neo offers centralized administration of cluster nodes comprising:

- ➔ Fully automated deployment of stateful and stateless operating systems on cluster nodes
- ➔ Parallel execution of administrative tasks on many or all cluster nodes
- ➔ A centralized administration environment to conduct common maintenance tasks
- ➔ Distribution and collection of files to and from cluster nodes
- ➔ Power management (on, off, reset) of cluster nodes
- ➔ Graphical (browser) or CLI-based console access to cluster nodes

In addition, LXC3-neo features tools for cluster users that facilitate using different environments for different compiler and library versions. These tools make it easy to switch between different versions of user applications or middleware like MPI, or switch the development tools from one version to another.

Graphical Administration Tool

LXC3-neo comprises a Graphical User Interface that can be used with a regular web browser, which eases daily administration tasks and provides an overview at a glance.



	Node Name	Power	Status	Group	VNFS	Description/Comments	Provision	Deploy	Debug
□	lxc001	●	ready	universe	centos6.stateless.x86_64		VNFS:centos6.stateless.x86_64	no	no
□	lxc002	●	ready	universe	centos6.stateless.x86_64		VNFS:centos6.stateless.x86_64	no	no
□	lxc003	●	[illegible]	universe	centos6.stateless.x86_64	maintenance	[power_off]	no	yes(1)
□	lxc004	●	ready	universe	centos6.stateless.x86_64		VNFS:centos6.stateless.x86_64	no	no
□	lxc005	●	init	universe	c6-torque4-2.stateful		VNFS:centos6.stateful-2	no	no
□	lxc006	●	init	universe	centos6.stateless.x86_64		[power_on]	no	no
□	lxc007	●	halt	universe	centos6.stateless.x86_64		VNFS:centos6.stateless.x86_64	no	no
□	lxc008	●	ready	universe	centos6.stateless.x86_64		VNFS:centos6.stateful-2	no	no
□	lxc009	●	ready	universe	centos6.stateless.x86_64		VNFS:centos6.stateless.x86_64	no	no
□	[illegible]	●	ready	universe	centos6.stateless.x86_64	interlagos in 2U twin chassis	VNFS: centos6.stateless.x86_64	no	yes(3)

Actions from the “Node Action” drop-down menu act on the marked (checked) nodes.

Supported Operating Systems

The following operating systems are supported on master nodes with full integration into the LXC3-neo framework when also deployed on compute nodes:
- Red Hat Enterprise Linux 7/8
- CentOS 7/8

The provisioning system can deploy any Linux operating system on compute nodes.

Resource Management

HPC clusters are shared among multiple users, and the coordination of resources is done with the help of a batch or queueing system which includes a resource scheduler. LXC3-neo clusters are pre-configured and ready to use with a resource management system based on either of the free and open source programs Torque/Maui or SLURM. The integration of other (commercial) resource management systems like PBS-pro, Torque/Moab or LSF is also possible.

The resource management system is setup with one default queue. The user interface consists of a set of command line utilities, which enables full control over jobs and their resource definitions.

Key features of the job scheduler:
- Backfill
- Fair Share Scheduling
- Topology awareness
- Job reservation
- Interactive jobs
- Job dependencies
- Job accounting
- Definition of own attributes
- and many more...

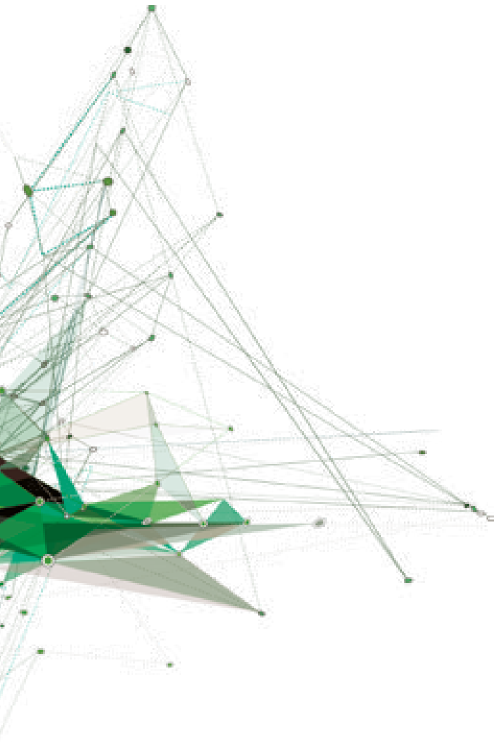
Other resource management systems (PBS-Pro, LSF, Grid Engine flavours) can be integrated upon request.

User Administration and System Integration

LXC3-neo can be integrated with widely used name services like LDAP or NIS, which makes user administration simple and fits into already installed management processes. However, when used with smaller clusters, LXC3-neo also works with the traditional UNIX user and group management tools.

Customer-specific Ansible Playbooks that are already in place can be integrated in order to configure the compute nodes. LXC3-neo's Ansible module pulls Playbooks from master nodes to achieve best scalability.

Integration and usage of customer site services like NTP, DNS and NFS shares is also possible.



Monitoring and Alerting System

LXC3-neo managed clusters are usually configured with performance and health monitoring systems.

Performance monitoring is handled by Ganglia, a well-known and commonly used scalable monitoring component originated by the University of Berkeley. Ganglia is configured to reflect the cluster monitoring hierarchy and provides a huge collection of internal metrics for each host which are propagated to the master node and stored as time series in round-robin databases. In addition NEC defined metrics can be fed in a simple way into the Ganglia infrastructure with the help of an LXC3-neo cluster metrics daemon.

Health monitoring of the cluster is done with the help of Nagios, a system health monitoring tool which watches hosts and services running on hosts. Nagios can send alerts to system administrators about problems and their recovery. Problems can trigger automatic actions like switching off nodes when the environment temperature is too high. Nagios can also be used to coordinate activities of multiple system administrators, e.g. by acknowledging problems, attaching comments to hosts or services or marking and allocating downtimes. The configuration of Nagios is rather complex therefore LXC3-neo clusters come preconfigured with a sensible default setup that partially uses the Ganglia performance monitoring infrastructure to transport measurement results for Nagios in a scalable way.

Popular monitoring and alerting stacks like ELK (Elasticsearch, Logstash and Kibana/Grafana) or TICK (Telegraf, InfluxDB, Chronograf and Kapacitor) with Grafana can be integrated. Using these tools usually requires a lot of changes to fit customer's expectations and site-specific requirements.

But regardless of which tools are used, the cluster can be monitored in an easy way, and in case it reaches a certain critical state that requires operator intervention, automatic console alerts and/or e-mail notifications can be created. Emergency actions like automatic shutdown of affected cluster nodes can be configured. Preset conditions for these actions are e.g.:

- Reaching a certain CPU temperature or fan speed
- Reaching a critical number of memory errors
- Faulty system components like HDD or power supply failure
- Complete node hardware failure
- Cluster over temperature

It is further possible to integrate monitoring and resource management.

Features

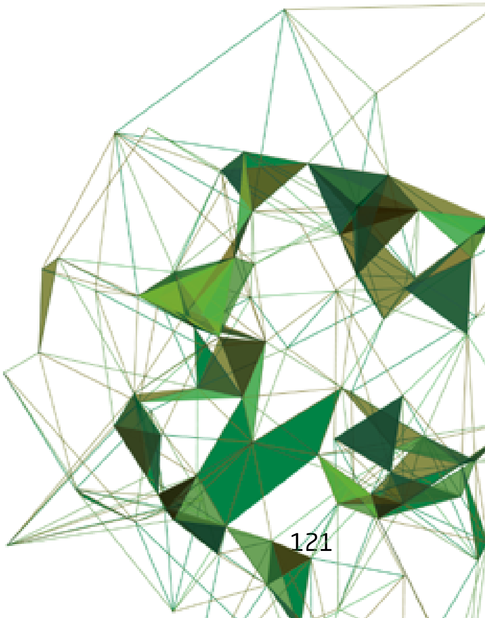
The following contains a list of checks and supervision provided in LXC3-neo. Most features are noted as check which implies automatic alert and notification via e-mail or some other facility.

- ✔ Check if nodes are running (compute and master nodes).
- ✔ Check node health of compute nodes (lbnl-nhc: Lawrence Berkeley National Labs Node Health Check) before and after running a job and periodically every 15 minutes.
- ✔ Automatically mark defective or not properly working compute nodes as down in the resource management system.
- ✔ Automatically add nodes that got repaired to the resource management system and make them available for batch jobs.
- ✔ Generate an alarm event (e-mail administrators or other) when a certain number of compute nodes are defective (not counting compute nodes that are marked down by administrators).
- ✔ Check date and time (DNS server) on master node(s).
- ✔ Check HTTP server on master node(s).

- ✔ Check status of High Availability system on master nodes.
- ✔ Check NFS (e.g. important for shared home directories) on master node(s).
- ✔ Check system disks (SMART status) on master node(s)
- ✔ Check swap activity on master and compute nodes.
- ✔ Prepared to hierarchically organize Nagios and push local check results to a (company/institution) centrally running Nagios.
- ✔ Check logfiles on master nodes for the following list of defects (this also covers compute nodes if system log is forwarded to the master node):
  - Kernel: I/O errors
  - Kernel: media errors
  - Kernel: ATA errors
  - Kernel: Call Trace
  - Kernel: Out Of Memory
  - Kernel: BUG: soft lockup
  - Kernel: CPU temperature above threshold
  - Kernel: CPU temperature above threshold, cpu clock throttled
  - Kernel: CPU package power limit notification exceeds given number of events
  - Kernel: Unhandled error code
  - Kernel: TCP RPC error
  - Kernel: Task blocked for more than n seconds
  - NFS: ARP neighbour table overflow
  - NFS: RPC error
  - NFS: peer name failed
  - Device mapper: multipath failed
  - Security: PAM request to sssd failed, connection refused
  - Security: Failed password for user root while login
  - Hardware: Processor heated above trip temperature, throttling enabled (MCE)
  - Hardware: Memory DIMMS ECC errors as correctable (warning) and uncorrectable (MCE)
  - Hardware: Power supply failure

  - Hardware: AC power lost
  - Hardware: Power supply temperature critical
  - System disks: SMART self check failure
  - Network: InfiniBand port hardware error
  - Storage: Fibre channel problems (abort command issued)
  - Storage: Lustre (LNet) read/write errors
  - Logrotate failed

- ✔ Immediately react on critical events like compute node overtemperature.
- ✔ Automatically feed Nagios check results as state metrics into Ganglia.
- ✔ Monitor and save performance and other metrics in a Round Robin Database. Metrics are:
  - Fan speed on master and compute nodes.
  - Default Ganglia (gmond) acquired metrics: boottime, bytes_in/bytes_out, cpu_aidle, cpu_idle, cpu_nice, cpu_num, cpu_speed, cpu_system, cpu_user, cpu_wio, disk_free, disk_total, load_fifteen, load_five, load_one, machine_type, mem_buffers, mem_cached, mem_free, mem_shared, mem_total, os_name, os_release, part_max_used, pkts_in, pkts_out, proc_run, proc_total, swap_free, swap_in, swap_out, swap_total
  - Nagios state metrics: nagios.Compute_Node_Queueing_State, nagios.Corosync_State, nagios.Dns_Server, nagios.HA_State, nagios.hoststate, nagios.Http_Server, nagios.Log_Files, nagios.Nfs_Server, nagios.Swap, nagios.System_Disks
  - Input power of compute nodes.
  - Power consumption per CPU (motherboard dependant).
- ✔ Simple Job monitoring by adding vertical bars in Ganglia charts at job start and end and assigning events to nodes on which a particular job is running (Torque and SLURM only).
- ✔ Command line tool to query state of Nagios checks (especially useful for remote administration where there is no graphical user interface available).
- ✔ Re-schedule Nagios checks via command line.
- ✔ Single node High Performance Linpack (HPL) is automatically run on compute nodes to check node health. Usually run once after boot but can be configured to run periodically as long as there is no job submitted.



- ✔ Well known STREAM memory benchmark is automatically run after boot to check memory and CPU health.
- ✔ Automatic issue creation in a remote bug tracking tool like Bugzilla or Jira in case a compute node fails health check and is taken out of the resource management system.
- ✔ Automatic collect and package most important configuration and log data in error cases to send as e-mail attachment to NEC customer support.
- ✔ And many more...

Services

Installation & Integration

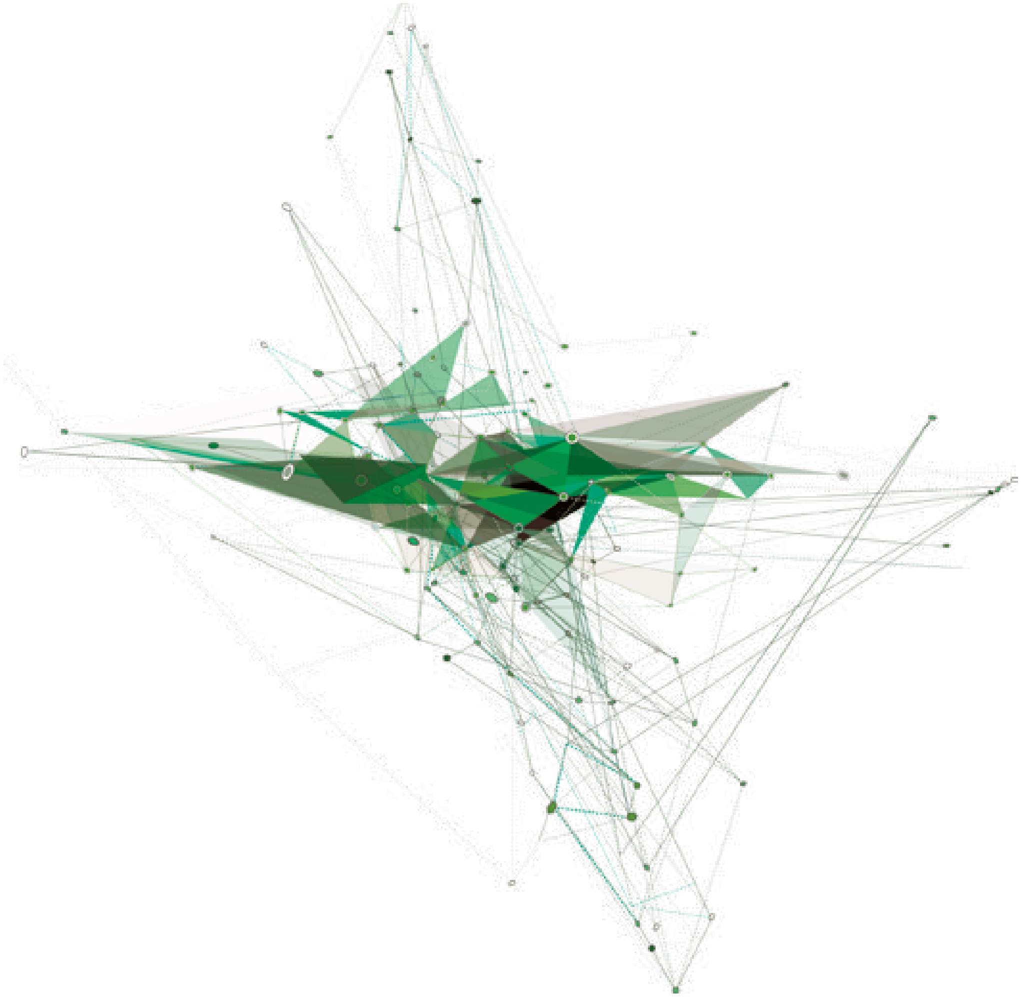
LXC3-neo offers fully automated installation of all cluster master nodes and an easy adaption to the customer's network environment and software infrastructure (see above).

Documentation and Training

NEC provides a detailed system description and documentation that explains system administration with LXC3-neo. In-depth trainings are also available.

Support

NEC provides a web-accessible ticketing system based on Request Tracker, which can be used for problem reporting and incident tracking. NEC implements ITIL-compliant procedures to standardize problem management and provide solutions in a timely manner.



Imprint

Text and Compilation:
Patrick Becker (patrick.becker@emea.nec.com)
Roberto Bifulco (roberto.bifulco@neclab.eu)
Rudolf Fischer (rudolf.fischer@emea.nec.com)
Erich Focht (erich.focht@emea.ec.com)
Andreas Jeutter (andreas.jeutter@emea.nec.com)
Alexander Menck (alexander.menck@emea.nec.com)
Shintaro Momose (shintaro.momose@emea.nec.com)
Oliver Tennert (oliver.tennert@emea.nec.com)

Concept:
Oliver Tennert (oliver.tennert@emea.nec.com)

Layout and Design:
neunpunktzwei Werbeagentur GmbH
www.neunpunktzwei.de

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

www.nec.com

HPC Division
Raiffeisenstr. 14
D-70771 Leinfelden-Echterdingen
Tel.: +49 (0) 711 78 055 0

HPC Division
3 Parc Ariane
F-78284 Guyancourt
Tel.: +33 (0) 139 30 66 00

HPCE UK Division
Unit 24, 29-30 Horse Fair
Banbury, Oxon, OX16 0BW
United Kingdom
DDI: +44 (0) 1295 814500

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

www.nec.com

HPC Division
Raiffeisenstr. 14
D-70771 Leinfelden-Echterdingen
Tel.: +49 (0) 711 78 055 0

HPC Division
3 Parc Ariane
F-78284 Guyancourt
Tel.: +33 (0) 139 30 66 00

HPCE UK Division
Unit 24, 29-30 Horse Fair
Banbury, Oxon, OX16 0BW
DDI: +44 (0) 1295 814500