

Use Intel® SSF to Remove Bottlenecks to AI

Patrick Mannion

Machine learning and AI are exposing the limitations of high-performance computing (HPC). In theory, the parallel architectures of HPC can scale linearly as resources are added—but in practice, this has not been the case.

HPC systems often encounter bottlenecks due to imbalances between processing, data transport, and storage. There is also a lack of standardization of hardware and software that has led to disjointed, suboptimal implementations. The result? As systems scale, so too do inefficiencies and costs.

The Emerging Scalable Systems Framework

To address the bottlenecks, Intel introduced the Intel® Scalable Systems Framework (Intel® SSF). SSF targets a more holistic, balanced, and scalable approach to HPC. To accomplish this, the SSF calls out specific elements around memory, processing, and network transport that work together with management software and supporting reference architectures (**Figure 1**).

Hardware elements that can be used within the SSF include:

- Intel® Xeon® processors E5-2600 v4 and Intel® Xeon Phi™ processors

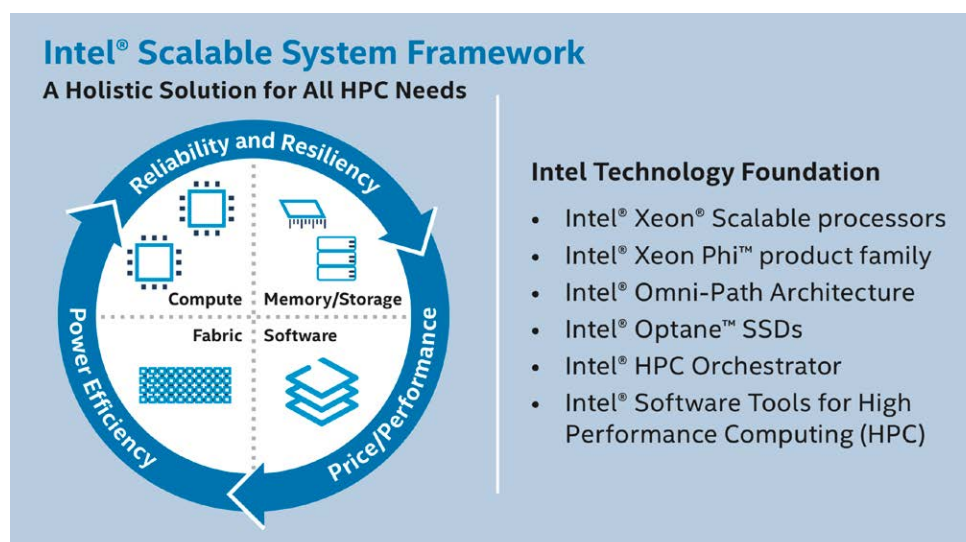


Figure 1. The SSF calls out specific elements around memory, processing, and network transport that work together with management software, and supporting reference architectures, to remove HPC bottlenecks. (Source: [Intel Corp.](https://www.intel.com))

- Intel® Optane™ SSDs built using NVMe (non-volatile memory express)
- Intel® Omni-Path Architecture (Intel® OPA) fabric and 10/40-Gbit/s Ethernet

These are supported by Intel® Enterprise Edition for Lustre* software (Intel® EE for Lustre software). Lustre is an open-source file system specifically designed to address the needs of parallel storage architectures. Intel builds on the popular file system with enhancements, including:

- Intel® Manager for Lustre, which simplifies install and configuration
- Integrated support for Hadoop* MapReduce*
- Global 24/7 technical support

Intel SSF also puts in place standards for operating systems, including the Linux kernel, access controls, programming interfaces, runtime environments, storage, and file systems, just to mention a few. To cite two examples, it specifies the Linux Standard Base (LSB) command system and the minimum amount of RAM on each node.

It's important to note that SSF uses the LP64 programming model for its APIs. That means it is compatible with common HPC programming models and can leverage existing code. This supports integration of various HPC capabilities that might have otherwise been disjointed.

The benefits of SSF are impressive. "With SSF you get to 25% to 30% faster than a previous [non-SSF-enabled] generation," said Andy Lee, server and storage product manager at [Premio Inc.](#), a provider of SSF-based storage solutions.

A practical example is self-driving cars, which are already generating terabytes of data as they evolve. "A previous-gen Xeon would take a month to analyze all the data and do the object training; now you can cut that time in half to train on all the objects for a self-driving car," said Lee.

Premio has already implemented SSF, using it as the foundation of its [FlacheSAN2N24U-D5](#) storage servers (**Figure 2**). The server uses two Intel® Xeon® scalable



Figure 2. The [FlacheSAN2N24U-D5](#) storage server from Premio Inc. uses the SSF to remove bottlenecks so it can reach to 60 Gbytes/s throughput, and 12 million IOPS. (Source: Premio Inc.)

processors and supports 24 front-accessible, hot-swap NVMe PCIe 3 x4 2.5-inch drives. Using the principles of SSF, and the other elements, such as Omni-Path and 100G interfaces, the FlacheSAN2N24U-D5 achieves a throughput of 60 Gbytes/s and 12 million IOPS.

The FlacheSAN2N24U-D5 is a supercomputing application, said Lee, "where you analyze the data quickly, such as for drilling, weather forecasting, oil and gas, agriculture, and security."

Not All SSF Implementations are the Same

While it may seem that the SSF makes it relatively easy to develop or select an SSF-based HPC, designers or potential customers need to be careful in their implementation or choice of vendor.

According to Lee, Premio's added value is in its implementation of SSF because it makes its own boards, does all the routing, and ties it directly to the drives for the storage (**Figure 3**). It also sources the right components for low latency and throughput, said Lee. But cost is still a factor, so Lee said they make a point of using off-the-shelf components.

Designing the HPC system is good, but needs to change and upgrades are always on the horizon. Premio addresses this directly. "The way we design our server, it's compatible with future processors," said Lee. "All you have to do is swap the

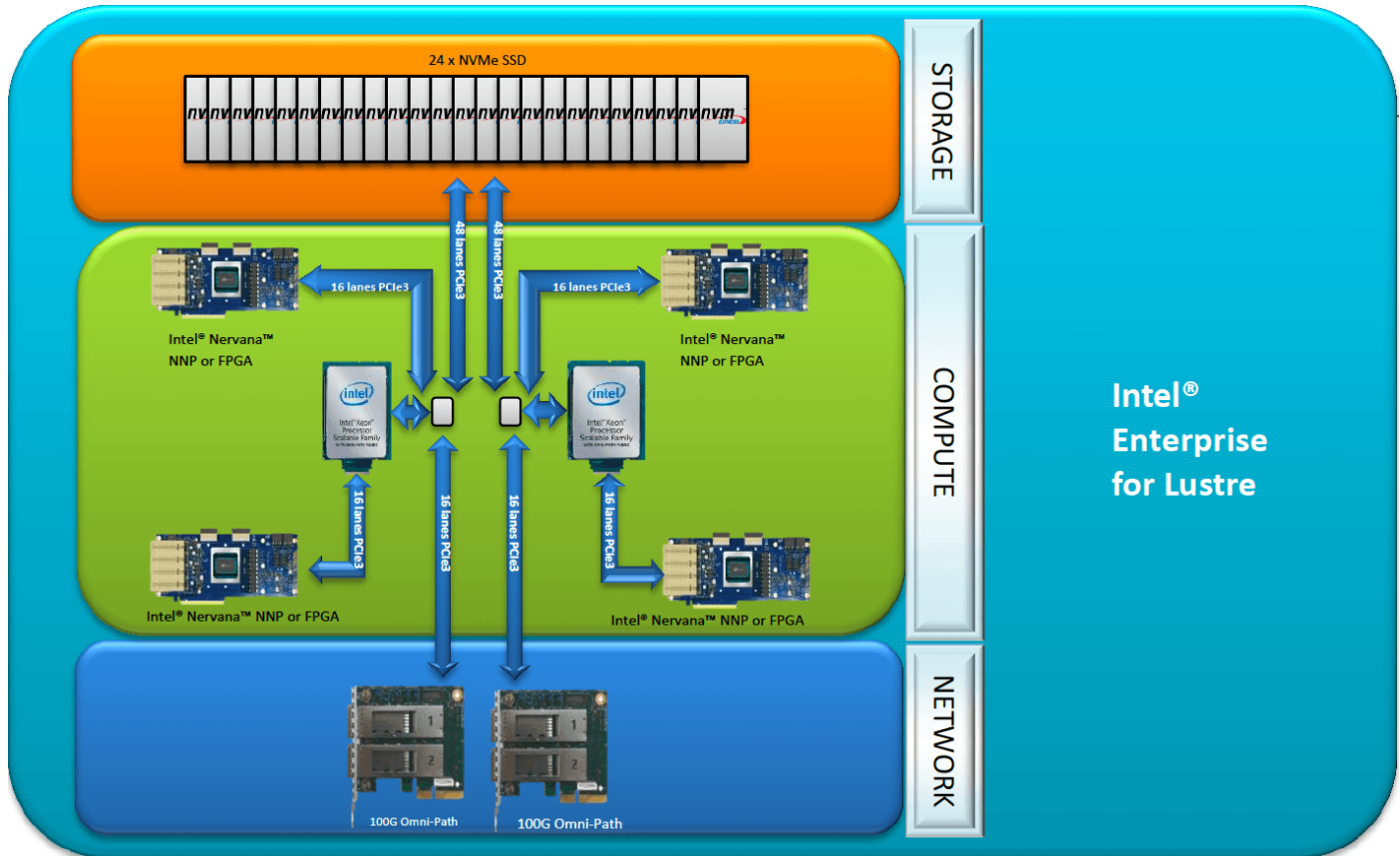


Figure 3. Premio differentiates its SSF implementation by doing its own board design and routing, and sourcing its own components for low latency and throughput. (Source: Premio Inc.)

[older] computing node and get Skylake [now called Xeon Scalable processors].” Using this modular, swap-in approach saves time, compared to redesigning a board, which Lee said could take from six months to a year.

The other caveat is that there’s no guarantee a design team will spend the time to architect the new SSF-based system correctly, said Lee, and it ends up with more bottlenecks. “We create a balanced architecture to eliminate all the bottlenecks through the network,” he said.

For example, Premio can take full advantage of the Xeon Scalable processor by being able to run all five PCIe lanes, where others may run only two. To manage the lanes, Premio also uses the RoCE (RDMA over converged Ethernet) network protocol. This is a link layer protocol, so it allows communication between any two hosts in the same Ethernet broadcast domain.

Premio has other SSF implementations, both available and in the works. With AI here and growing fast, the timing couldn’t be better.